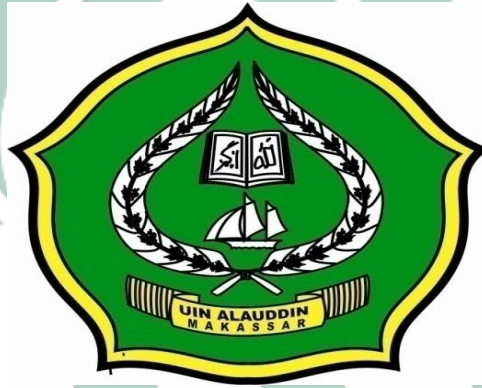


**APLIKASI ALGORITMA BELLMAN-FORD DALAM
MEMINIMUMKAN RUTE PERJALANAN TUKANG BENTOR
DI KECAMATAN BIRINGKANAYA**



SKRIPSI

Diajukan untuk Memenuhi Salah Satu Syarat Meraih Gelar Sarjana Sains (S.Si)

Jurusan Matematika pada Fakultas Sains dan Teknologi

UIN Alauddin Makassar

RASDIANA

60600110038

JURUSAN MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI ALAUDDIN

MAKASSAR

2015

PENGESAHAN SKRIPSI

Skripsi yang berjudul “Aplikasi Algoritma Bellman-Ford dalam Meminimumkan Rute Perjalanan Tukang Bentor di Kecamatan Biringkanaya”, yang disusun oleh saudari **RASDIANA**, Nim: **60600110038** Mahasiswa Jurusan Matematika pada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, telah diuji dan dipertahankan dalam sidang *munaqasyah* yang diselenggarakan pada hari Kamis tanggal **03 September 2015 M**, bertepatan dengan **19 Dzulkaidah 1436 H**, dinyatakan telah dapat diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains (S.Si.).

Makassar, 03 September 2015 M
19 Dzulkaidah 1436 H

DEWAN PENGUJI

Ketua	: Prof. Dr. H. Arifuddin Ahmad, M.Ag	(.....)
Sekretaris	: Ermawati, S.Pd., M.Si.	(.....)
Munaqisy I	: Wahyuni Abidin, S.Pd., M. Pd.	(.....)
Munaqisy II	: Arifin, S.Si., M.Si.	(.....)
Munaqisy III	: Dr. Hasyim Haddade, S.Ag., M.Ag.	(.....)
Pembimbing I	: Irwan, S.Si., M.Si.	(.....)
Pembimbing II	: Bahtiar, S.Pd., M. Si.	(.....)

Diketahui oleh:

Dekan Fakultas Sains dan Teknologi
UIN Alauddin Makassar

Prof. Dr. H. Arifuddin Ahmad, M.Ag
Nip. 19691205 199303 1 001

SURAT PERNYATAAN KEASLIAN SKRIPSI

saya yang bertanda tangan di bawah ini:

Nama : Rasdiana

NIM : 60600110038

Jurusan/Fakultas : Matematika/ Sains dan Teknologi

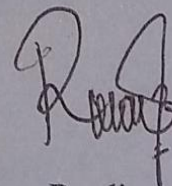
Judul Skripsi : Aplikasi Algoritma Bellman-Ford Dalam
Meminimumkan Rute Perjalanan Tukang Bentor di
Kecamatan Biringkanaya.

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peraturan yang berlaku.

Makassar, September 2015

Yang membuat pernyataan



Rasdiana

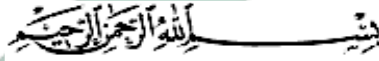
MOTTO

“Kesulitanmu itu sementara, seperti semua sebelumnya yang pernah terjadi maka tetap lah berusaha dan berdoa namun Janganlah meminta bukti bahwa doamu akan dijawab oleh Tuhan, tapi buktikanlah kesungguhan dari usaha dan doamu”



UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R

KATA PENGANTAR



Alhamdulillah rabbil'alam, segala puji syukur kehadiran Allah Swt atas limpahan rahmat, taufiq dan hidayah-Nya, sehingga penulis mampu menyelesaikan penulisan skripsi yang berjudul “Aplikasi Bellman-Ford Dalam Meminimumkan Rute Perjalanan Tukang Bentor Di Kecamatan Biringkanaya”. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan Nabi besar Muhammad Saw., sebagai *uswatun hasanah* dalam meraih kesuksesan di dunia dan akhirat.

Melalui tulisan ini pula, penulis menyampaikan ucapan terima kasih yang tulus, teristimewa kepada kedua orang tua tercinta Ayahanda **Hambali** dan Ibunda **Husna** atas segala do'a, restu, kasih sayang, pengorbanan dan perjuangan yang telah diberikan selama ini serta telah memberikan dukungan dan doa kepada penulis dan menjadi motivasi terbesar bagi penulis untuk segera menyelesaikan studi. Kepada beliau penulis senantiasa memanjatkan do'a semoga Allah Swt., mengasihi dan mengampuni dosanya. Amin. Untuk Saudara-saudaraku **Wahidah**, **Nurhasna**, **Faisal** dan **Ika Rezky** tercinta. Untuk Nenekku **Raesa** dan kakekku **Abd. Hamid tersayang** serta keluarga besarku yang selalu memberikan do'a, semangat dan dukungan selama ini.

Keberhasilan penulisan skripsi ini tidak lepas dari bimbingan, pengarahan dan bantuan dari berbagai pihak baik berupa pikiran, motivasi, tenaga, maupun do'a. Karena itu penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. H. Musafir Pababbari, M.Si, Rektor UIN Alauddin Makassar beserta seluruh jajarannya.
2. Bapak Prof. Dr. H. Arifuddin, M.ag, Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar.
3. Bapak Irwan, S.Si.,M.Si. dan Ibu Wahidah Alwi, S.Si.,M.Si, ketua dan sekretaris Jurusan Matematika
4. Bapak Irwan, S.Si.,M.Si. dan Bapak Bahtiar, S.Pd.,M.Si, pembimbing I dan II yang dengan sabar telah meluangkan waktu demi memberikan bimbingan dan pengarahan dalam penyelesaian skripsi ini.
5. Seluruh dosen jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar yang telah menyalurkan ilmunya kepada penulis selama berada di bangku kuliah.
6. Segenap karyawan dan karyawan Fakultas Sains dan Teknologi yang telah bersedia melayani penulis dari segi administrasi dengan baik selama penulis terdaftar sebagai mahasiswa Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar.
7. Seluruh teman-teman seperjuangan di keluarga **AKSIOMA 010** terkhusus untuk sahabat-sahabat **terdekatku(Uni, Idha, Ria dan Lia)** serta teman-teman **COLAPS 010** (Yaya, Nuni, Dhila, Jannah, ati, vivi, sri, ani, adi, ciwan, tuti, chuwa, uni, supi, ida, dan ayu) yang telah memotivasi penulis untuk segera menyelesaikan studi serta dukungan dan canda tawa yang menyisakan kesan mendalam di hati.

8. Kanda-kanda senior dan adik-adikku, serta seluruh warga **HMJ MATEMATIKA** sebagai keluarga keduaku atas pengalaman dan nasehat-nasehatnya sehingga penulis dapat lebih mengerti arti pentingnya kebersamaan.
9. Sahabat-sahabatku **Uni, Anca, Muhlis, Imah, Sidar, Kak Anto, Kahar, Isal, dan Akir** yang tak pernah bosan mengingatkan dan memberi semangat dalam menjalani masa-masa kuliah.
10. Teman-teman seperjuanganku selama **KKN Regular 49** di Desa Bontolangkasa Selatan, Kec.Bontonompo, Kab.Gowa (Ijha, Sari, Tuti, Anwar, Febry, Irsan, Bacci dan Ippank) yang selalu memberi semangat dalam menjalani proses ini.
11. Teman-teman yang telah membantu penulis dalam proses pengambilan data di lapangan dan telah banyak memberikan bantuan berupa moril dan materil yang tidak bisa saya sebutkan namanya satu persatu. Rasa terima kasih yang tiada hentinya penulis haturkan, semoga bantuan yang telah diberikan bernilai ibadah di sisi Allah Swt., dan mendapat pahala yang setimpal. Amin.

Akhirnya, diharapkan agar hasil penelitian ini dapat bermanfaat dan menambah khasanah ilmu pengetahuan.

Amin Ya Rabbal Alamin

Makassar, Desember 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	
PENGESAHAN SKRIPSI	i
SURAT PERNYATAAN KEASLIAN SKRIPSI	ii
MOTTO	iii
KATA PENGANTAR	iv
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
ABSTRAK	xi
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	7
C. Tujuan Penelitian.....	7
D. Batasan Masalah	8
E. Manfaat Penelitian.....	8
F. Sistematika Penulisan	8
BAB II KAJIAN PUSTAKA	10
A. Teori Graf	10
B. Lintasan Terpendek	19
C. Algoritma Bellman-Ford	21
D. Matlab.....	27
BAB III METODOLOGI PENELITIAN	30
A. Jenis Penelitian	30

B. Waktu dan Lokasi Penelitian	30
C. Prosedur Penelitian	30
BAB IV HASIL DAN PEMBAHASAN	32
A. Hasil Penelitian.....	32
B. Pembahasan	50
BAB V PENUTUP	66
A. Kesimpulan.....	66
B. Saran.....	67
DAFTAR PUSTAKA	
LAMPIRAN	

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
 M A K A S S A R

DAFTAR GAMBAR

Gambar 2.1 Graf 5 titik dan 6 sisi	10
Gambar 2.2 Graf dengan 6 titik dan 10 sisi	11
Gambar 2.3 Graf komplit dengan 4 titik dan Graf komplit dengan 5 titik ...	12
Gambar 2.4 Graf Kosong dengan 3 titik	13
Gambar 2.5 Graf G Bifartisi.....	14
Gambar 2.6 Graf $K_{3,2}$ Bipartisi komplit	14
Gambar 2.7 Graf tak berarah.....	15
Gambar 2.8 Graf Berarah.....	16
Gambar 2.9 Graf G.....	17
Gambar 2.10 Graf terhubung	19
Gambar 2.11 Graf bebobot negatif.....	24
Gambar 2.12 Tahap pertama Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11	25
Gambar 2.13 Tahap kedua Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11	25
Gambar 2.14 Tahap ketiga Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11	25
Gambar 2.15 keempat Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11	26
Gambar 2.16 Tahap kelima Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.12	26

Gambar 2.17 Lintasan terpendek untuk penyelesaian contoh graf pada gambar 2.12 sebesar -1	26
Gambar 4.1 Peta Google Maps Kecamatan Biringkanaya	32
Gambar 4.2 Graf Rute jalan kecamatan Biringkanaya	32
Gambar 4.3 Tahap pertama Algoritma Bellman-Ford	36
Gambar 4.4 Tahap kedua Algoritma Bellman-Ford	37
Gambar 4.5 Tahap ketiga Algoritma Bellman-Ford (iterasi pertama)	38
Gambar 4.6 Tahap ketiga Algoritma Bellman-Ford (iterasi kedua)	39
Gambar 4.7 Tahap ketiga Algoritma Bellman-Ford (iterasi ketiga)	40
Gambar 4.8 Tahap ketiga Algoritma Bellman-Ford (iterasi keempat)	41
Gambar 4.9 Tahap ketiga Algoritma Bellman-Ford (iterasi kelima)	42
Gambar 4.10 Tahap ketiga Algoritma Bellman-Ford (iterasi keenam)	43
Gambar 4.11 Tahap ketiga Algoritma Bellman-Ford (iterasi ketujuh)	44
Gambar 4.12 Tahap ketiga Algoritma Bellman-Ford (iterasi kedelapan)	45
Gambar 4.13 Tahap ketiga Algoritma Bellman-Ford (iterasi kesembilan)	46
Gambar 4.14 Tahap ketiga Algoritma Bellman-Ford (iterasi kesepuluh)	47

ABSTRAK

Nama :Rasdiana
Nim :60600110038
Jurusan :Matematika
Judul :”Aplikasi Algoritma Bellman – ford dalam Meminimumkan Rute Perjalanan Tukang Bentor di Kecamatan Biringkanaya”

Algoritma merupakan kumpulan perintah-perintah untuk menyelesaikan suatu masalah tertentu. pada penelitian ini digunakan Algoritma Bellman-Ford, untuk menentukan lintasan terpendek untuk 31 titik batas jalan yang terdapat di 15 nama jalan yang ada di Salah satu wilayah kota Makassar tepatnya di kecamatan Biringkanaya terhadap rute perjalanan Tukang Bentor.

Untuk pencarian jalur perjalanan Tukang Bentor dilakukan dengan cara manual dan menggunakan *software MATLAB*. Dan pada bagian tugas akhir ini disimpulkan jarak minimum yang ditempuh Tukang Bentor dari tempat pangkalan Bentor(titik 1) ke semua titik batas jalan menggunakan Algoritma Bellman-Ford berdasarkan data yang didapatkan adalah Dari titik 1→2 sepanjang 2.100 meter, dari titik 1→3 sepanjang 3.100 meter, dari titik 1→4 sepanjang 5.960 meter, dari titik 1→5 sepanjang 6. 547 meter, dari titik 1→6 sepanjang 6.520 meter, dari titik 1→7 sepanjang 6.248 meter, dari titik 1→8 sepanjang 6.795 meter, dari titik 1→9 sepanjang 3.130 meter, dari titik 1→10 sepanjang 3.505 meter, dari titik 1→11 sepanjang 4.940 meter, dari titik 1→12 sepanjang 5.675 meter, dari titik 1→13 sepanjang 6.725 meter, dari titik 1→14 sepanjang 7.945 meter, dari titik 1→15 sepanjang 7.740 meter, dari titik 1→16 sepanjang 7.053 meter, dari titik 1→17 sepanjang 9.171 meter, dari titik 1→18 sepanjang 7.775 meter, dari titik 1→19 sepanjang 7.207 meter, dari titik 1→20 sepanjang 7.642 meter, dari titik 1→21 sepanjang 6.655 meter, dari titik 1→22 sepanjang 7.075 meter, dari titik 1→23 sepanjang 6.025 meter, dari titik 1→24 sepanjang 8.405 meter, dari titik 1→25 sepanjang 6.865 meter, dari titik 1→26 sepanjang 7.110 meter, dari titik 1→27 sepanjang 8.370 meter, dari titik 1→28 sepanjang 7.135 meter, dari titik 1→29 sepanjang 7.705 meter, dari titik 1→30 sepanjang 7.880 meter, dari titik 1→31 sepanjang 8.615 meter.

Kata Kunci: Algoritma Bellman-Ford, Lintasan tependek.

BAB I

PENDAHULUAN

A. Latar Belakang

Transportasi merupakan bidang kegiatan yang sangat penting dalam kehidupan masyarakat Indonesia. Keberadaan transportasi tidak lain adalah sebagai penunjang aktifitas manusia sehari-hari, dan merupakan sarana mobilitas di darat, laut dan udara. Di Indonesia transportasi selalu mengalami perkembangan dari masa ke masa seiring dengan laju perkembangan dunia saat ini. Peradaban manusia dan pengaruh kemajuan teknologi menjadikan transportasi berkembang kian modern. Pengaruh industrialisasi yang identik dengan penggunaan mesin dalam berbagai bidang kehidupan, mempengaruhi pula dalam perkembangan dunia transportasi. Manusia mulai menciptakan sarana transportasi bermesin seperti pesawat, mobil, motor, maupun kereta api. Seiring dengan perkembangan ini, maka sarana transportasi modern mulai menggantikan sarana transportasi tradisional yang telah lebih dulu dikenal.¹ Hal ini bersesuaian dengan firman Allah dalam QS. Yunus ayat 101:

قُلْ أَنْظَرُوا مَاذَا فِي السَّمَوَاتِ وَالْأَرْضِ وَمَا تُغْنِي الْآيَاتُ وَالنُّذُرُ عَنْ قَوْمٍ لَا يُؤْمِنُونَ ﴿١٠١﴾

¹ Stepen K. Sanderson, *Makro Sosiologi: Sebuah Pendekatan Terhadap Realitas Sosial*, (jakarta: Rajawali Pers, 2003), h. 64.

Terjemahnya:

Katakanlah: "Perhatikanlah apa yaag ada di langit dan di bumi. tidaklah bermanfaat tanda kekuasaan Allah dan rasul-rasul yang memberi peringatan bagi orang-orang yang tidak beriman".²

Melalui ayat tersebut Allah swt. menjelaskan bahwa Allah tidak akan memaksa, engkau tidak perlu memaksa mereka agar beriman, tetapi *katakanlah* kepada mereka, "Perhatikan dengan mata kepala dan hati masing-masing *apa*, yakni makhluk dan atau sistem kerja yang ada di langit dan di bumi. Sungguh banyak yang dapat kamu perhatikan, satu diantaranya saja bila kamu menggunakan akalmu yang dianugerahkan Allah swt. sudah cukup untuk mengantar kamu beriman dan menyadari bahwa Allah Mahakuasa.³

Ayat di atas merupakan ayat yang memberikan motivasi agar manusia ingin berkarya dengan menggunakan akal dan fikiranya. Dan salah contohnya yaitu Bentor, yang pada awalnya merupakan Becak yang termasuk ke dalam salah satu alat transportasi darat yang masih tradisional. Namun, akibat kemajuan pola pikir manusia kendaraan tradisional ini mulai mengalami inovasi ke arah yang lebih modern, efisien dan praktis.⁴

² Departemen Agama RI, *Alqur'an dan Terjemahannya*, (Jakarta: Perwakilan Bagian Percetakan dan Penerbitan Kementerian Agama, 2002), h. 220

³ M. Quraish Shihab, *Tafsir Al-Misbah: Pesan, kesan dan Keserasian Al-Qur'an*, (Jakarta: Lentera hati, 2002), h. 515.

⁴ Doyle Paul Johnson, *Teori sosiologi Klasik dan Modern*, (Jakarta; Gramedia, 1988), h. 220.

Bentor sekarang merupakan salah satu alat transportasi yang banyak digunakan, terutama orang-orang yang tempat tinggalnya jauh dari jalan raya karena bentor merupakan salah satu kendaraan yang tidak ditentukan jalurnya, sehingga dapat mengantar penumpang langsung ke tempat tujuan. Selain itu dengan menggunakan bentor kita dapat diantar langsung ke tempat tujuan tanpa menunggu penumpang lain seperti angkutan umum. Namun, salah satu yang sering muncul dalam kehidupan nyata yaitu masalah transportasi dari suatu tempat ke tempat yang lain dengan jarak atau biaya minimum. Sehingga pada tugas akhir ini penulis melakukan penelitian mengenai jalur terpendek.

Pemilihan jalur terpendek salah satu tujuan utamanya adalah untuk memudahkan aktivitas manusia.

Sebagaimana firman Allah dalam potongan ayat QS. Al-Baqarah ayat 185:

يُرِيدُ اللَّهُ بِكُمُ الْيُسْرَ وَلَا يُرِيدُ بِكُمُ الْعُسْرَ

Terjemahnya :

“Allah menghendaki kemudahan bagimu dan tidak menghendaki kesukaran bagimu”.⁵

Dalam tafsir ibnu katsir dikatakan bahwa orang yang sedang sakit atau dalam bepergian jauh bolehlah mereka berbuka, karena Allah stw. tidak menghendaki kesukaran bagi umatnya, tapi Allah menghendaki

⁵ Departemen Agama RI, *Alqur'an dan Terjemahannya*, h. 28.

kemudahan bagi hamba-Nya.⁶ Hal ini juga berkaitan dengan pemilihan lintasan terpendek, dimana dengan memilih lintasan terpendek akan menjadikan pekerjaan manusia mudah, namun harus tetap mengikuti peraturan yang berlaku.

Dalam pencarian rute terpendek pada suatu masalah terdapat banyak algoritma yang dapat digunakan. Pemilihan algoritma yang optimum selalu menjadi permasalahan dalam pencarian rute terpendek, dimana masing-masing algoritma memiliki kekurangan dan kelebihan tersendiri. Algoritma yang dapat digunakan untuk menentukan rute terpendek seperti algoritma Dijkstra, algoritma Floy-Warshall, algoritma Fleury, algoritma Greedy, algoritma Floyd dan algoritma Bellman-Ford. Penemu algoritma adalah tokoh muslim, berikut ada beberapa ilmuwan muslim penemu di bidang matematika yang lainnya yaitu sebagai berikut.

1. Al-Khawarizmi (Khawarizm, Uzbekistan, 194 H/780 M-Baghdad, 266 H/850 M). Ilmuwan muslim, ahli di bidang ilmu matematika, astronomi, dan geografi. Nama lengkapnya adalah Abu Ja'far Muhammad bin Musa Al-Khawarizmi dan di barat ia lebih dikenal dengan nama Algoarisme atau algorisme Dalam bukunya Al-Khawarizmi memperkenalkan kepada dunia ilmu pengetahuan angka 0 (nol) yang dalam bahasa arab disebut sifr
2. Al-Karaji dianggap sebagai ahli matematika terkemuka dan pandang sebagai orang pertama yang membebaskan aljabar dari operasi

⁶ Ibnu Katsir Ad-Dimayqi, *Tafsir Ibnu Katsir Juz I*, (Bandung: Penerbit Sinar baru Algensindo, 2000), h. 350.

geometris yang merupakan produk aritmatika Yunani dan menggantinya dengan jenis operasi yang merupakan inti dari aljabar pada saat ini. Karyanya pada aljabar dan polynomial memberikan aturan pada operasi aritmatika untuk memanipulasi polynomial

3. Al-Battani atau Muhammad Ibn Jabir Ibn Sinan Abu Abdullah dikenal sebagai bapak trigonometri. Ia lahir di Battan, Mesopotamia, dan meninggal di Damaskus pada tahun 929. Al-Battani adalah tokoh bangsa Arab dan gubernur Syria. Dia merupakan astronom Muslim terbesar dan ahli matematika ternama. Al-Battani melahirkan trigonometri untuk level lebih tinggi dan orang pertama yang menyusun tabel cotangen.
4. Umar Khayyam memiliki kontribusi besar dalam bidang matematika, terutama dalam bidang aljabar dan trigonometri. Ia merupakan matematikawan pertama yang menemukan metode umum penguraian akar-akar bilangan tingkat tinggi dalam aljabar, dan memperkenalkan solusi persamaan kubus.
5. Al-Qalasadi dalam mengembangkan matematika sungguh sangat tak ternilai. Ia sang matematikus Muslim di abad ke-15, kalau tanpa dia boleh jadi dunia dunia tak mengenal simbol-simbol ilmu hitung. Sejarah mencatat, al Qalasadi merupakan salah seorang matematikus Muslim yang berjasa memperkenalkan simbol-simbol Aljabar. Symbol-simbol tersebut pertama kali dikembangkan pada abad 14 oleh Ibnu al-Banna kemudian pada abad 15 dikembangkan oleh al-Qalasadi,

al-Qalasadi memperkenalkan symbol-simbol matematika dengan menggunakan karakter dari alphabet Arab.⁷

Terdapat beberapa penelitian tentang pencarian jalur terpendek diantaranya:

1. Raden Aprian Diaz Novandi meneliti tentang perbandingan algoritma Dijkstra dan algoritma Floyd-Warshall dalam penentuan lintasan terpendek pada tahun 2007.
2. Tahun 2009 penelitian Sri Handika tentang Algoritma Bellman-Ford sebagai solusi akses tercepat dalam jaringan komputer.
3. Tahun 2012 perancangan aplikasi untuk menentukan jalur terpendek menggunakan Algoritma Floyd di wilayah Purbalingga yang diteliti oleh Irfan Ardiansyah, dkk.

Pada penelitian ini menggunakan algoritma Bellman-Ford untuk meminimumkan rute perjalanan Tukang Bentor. Algoritma Bellman-Ford menghitung jarak terpendek (dari satu sumber) pada sebuah digraf berbobot. Maksudnya dari satu sumber ialah bahwa ia menghitung semua jarak terpendek yang berawal dari satu titik asal. Algoritma Bellman-ford mempunyai keistimewaan dibandingkan dengan algoritma-algoritma yang

⁷ Anonim, <http://repository.usu.ac.id/bitstream/123456789/42368/5/Chapter%20I.pdf> (Mei 2014).

lain, yaitu dapat digunakan untuk menyelesaikan permasalahan yang di dalamnya terdapat bobot (biaya) bernilai negatif.⁸

Berdasarkan latar belakang di atas, penulis mengangkat permasalahan tentang **“Aplikasi Algoritma Bellman – Ford dalam Meminimumkan Rute Perjalanan Tukang Bentor di Kecamatan Biringkanaya”**

B. Rumusan Masalah

Berdasarkan latar belakang di atas, maka didapat rumusan masalah yaitu bagaimana menentukan lintasan minimum yang ditempuh Tukang Bentor yang ada di kecamatan Biringkanaya dengan menggunakan *Algoritma Bellman- Ford*?

C. Tujuan Penelitian

Adapun tujuan yang ingin dicapai yaitu untuk mengetahui lintasan minimum yang ditempuh Tukang Bentor yang ada di kecamatan Biringkanaya dengan menggunakan *Algoritma Bellman- Ford*.

⁸ Andana, Galih, *Algoritma Bellman – Ford dalam distance Vektor Routing Protocol*”, <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2009-2010/Makalah2009/MakalahIF3051-2009-044.pdf>, 2009, (25 Mei 2014).

D. Batasan Masalah

Pada penyusunan skripsi ini akan dibahas Aplikasi Bellman – Ford dalam menentukan lintasan minimum tukang bentor yang ada di kecamatan Biringkanaya. Pembahasan skripsi ini dibatasi pada aplikasi algoritma Bellman – Ford dan hanya 15 jalan yang diteliti di kecamatan Biringkanaya.

E. Manfaat Penelitian

1. Bagi penulis, menambah dan memperluas pengetahuan tentang algoritma Bellman – Ford dalam kehidupan nyata.
2. Bagi pembaca, sebagai bahan pertimbangan dalam pengambilan keputusan tentang meminimumkan rute perjalanan tukang bentor.
3. Bagi lembaga UIN Alauddin Makassar, Sebagai bahan kepustakaan yang dijadikan sarana pengembangan wawasan keilmuan, khususnya di Jurusan Matematika.
4. Bagi pengguna Bentor, untuk mempermudah perjalanan jika ingin bepergian.

F. Sistematika Penulisan

Agar penulisan tugas akhir ini tersusun secara sistematis, maka penulis memberikan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan meliputi : latar belakang permasalahan, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bagian ini terdiri atas konsep-konsep (teori-teori) yang mendukung bagian pembahasan. Konsep-konsep tersebut antara lain membahas tentang graf dan Algoritma Bellman – ford.

BAB III METODE PENELITIAN

Dalam bab ini dikemukakan metode yang berisi langkah-langkah yang ditempuh untuk memecahkan masalah, yaitu jenis penelitian, sumber data, waktu dan lokasi penelitian dan prosedur penelitian.

BAB II

TINJAUAN PUSTAKA

A. Teori Graf

1. Pengertian Graf

Definisi 2.1. Graf G didefinisikan sebagai pasangan himpunan (V, E) yang dalam hal ini, V adalah himpunan tidak kosong dari titik-titik dan E adalah himpunan sisi yang menghubungkan sepasang titik.

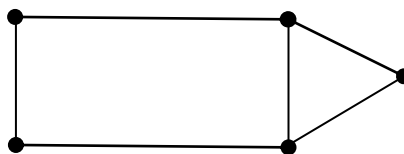
$$V = \{v_1, v_2, \dots, v_n\} \quad (1)$$

$$E = \{e_1, e_2, \dots, e_n\} \quad (2)$$

Atau dapat ditulis singkat

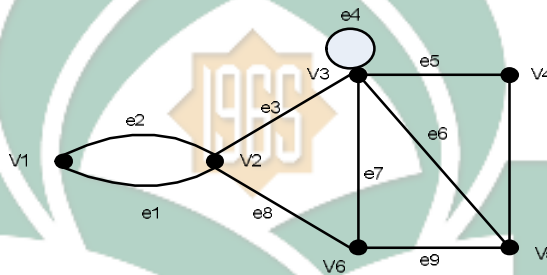
$$G = (V, E) \quad (3)$$

Secara umum graf dapat digambarkan dengan suatu diagram dengan titik ditunjukkan sebagai titik yang dinotasikan dengan v_i , $i = 1, 2, \dots, n$ dan sisi digambarkan dengan sebuah garis lurus atau garis lengkung yang menghubungkan dua buah titik (v_i, v_j) dan dinotasikan dengan e_k .



Gambar 2.1. Graf 5 titik dan 6 sisi

Definisi 2.2. Loop adalah sisi yang berawal dan berakhir pada titik yang sama, sedangkan sisi paralel adalah dua sisi atau lebih berbeda yang menghubungkan dua buah titik v_i dan v_j yang sama.



Gambar 2.2. Graf dengan 6 titik dan 10 sisi

Pada Gambar 2.2 dapat dilihat bahwa e_4 adalah sebuah loop dan e_1 serta e_2 adalah dua buah sisi yang paralel.

Definisi 2.3. Graf sederhana adalah graf yang tidak memuat loop dan sisi-sisi paralel. Misalkan $V = (v_1, v_2, v_3, v_4, v_5)$ dan $E = (e_1, e_2, e_3, e_4, e_5, e_6)$, maka $G = (V, E)$ adalah graf sederhana yang dapat dilihat pada Gambar 2.1.

Definisi 2.4. Suatu sisi e_k dalam suatu graf G dengan titik-titik ujung v_i dan v_j disebut saling incident dengan v_i dan v_j , sedangkan v_i dan v_j ini disebut dua buah titik yang saling adjacent. Jika kedua sisi tersebut incident pada suatu titik persekutuan, maka dua buah sisi e_k dan e_m disebut saling adjacent. Pada Gambar 2.2 dapat dilihat bahwa e_8, e_7, e_9 adalah tiga buah sisi yang incident dengan v_6 , sedangkan e_5 dan e_7 adalah adjacent.

Definisi 2.5. Derajat dari sebuah titik v_i dalam graf G adalah jumlah sisi yang incident dengan v_i dengan loop dihitung dua kali. derajat dari sebuah titik v_i biasanya dinotasikan dengan $d(v_i)$.

Pada Gambar 2.2 dapat dilihat bahwa $d(v_1) = 2$, $d(v_2) = 4$, $d(v_3) = 5$, $d(v_4) = 2$, $d(v_5) = 2$, $d(v_6) = 3$, dan $d(v_7) = 3$.

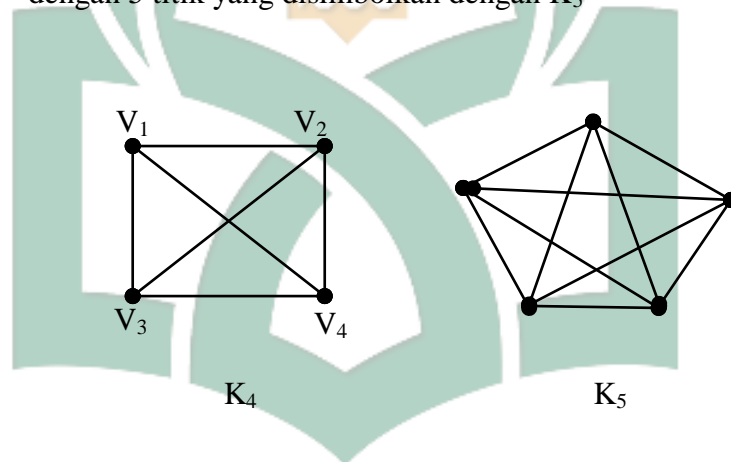
Definisi 2.6. Suatu walk dalam sebuah graf $G(V,E)$ adalah suatu barisan berhingga dari titik dan sisi secara bergantian yang dimulai dan diakhiri dengan titik sehingga setiap sisi incident dengan titik sebelum dan sesudahnya, di mana sebuah sisi hanya dilalui satu kali. Di dalam suatu walk pada sebuah graf dapat terjadi bahwa satu titik dilalui lebih dari satu kali.

Definisi 2.7. Suatu Graf Berarah G terdiri dari himpunan titik $V(G)$: $\{v_1, v_2, \dots\}$, himpunan sisi $E(G)$: $\{e_1, e_2, \dots\}$, dan suatu fungsi ψ yang menghubungkan setiap sisi dalam $E(G)$ ke suatu pasangan berurutan titik (v_i, v_j) . Jika $e_k = (v_i, v_j)$ adalah suatu sisi dalam G , maka v_i disebut titik awal e_k dan v_j disebut titik akhir e_k . Arah sisi adalah dari v_i ke v_j .⁹

⁹Munir, Rinaldi, "Matematika Diskrit Program Studi Informatika", Institute Teknologi bandung, <http://id.wikipedia.org>. Makalah IF2091, 2009.

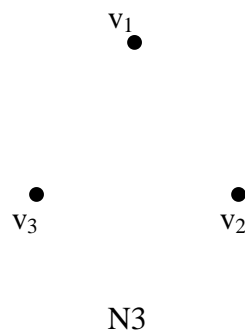
2. Jenis-jenis Graf

- a. Sebuah graf komplit (graf lengkap) dengan n titik, dilambangkan dengan K_n adalah graf sederhana dengan n titik dan setiap dua titik berbeda dihubungkan dengan sebuah sisi. Misalkan graf komplit dengan 4 titik yang disimbolkan dengan K_4 dan graf komplit dengan 5 titik yang disimbolkan dengan K_5



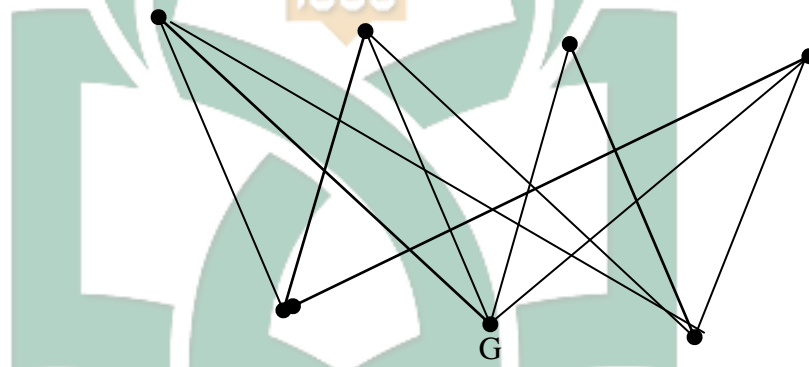
Gambar 2.3 K_4 graf komplit dengan 4 titik K_5 graf komplit dengan 5 titik

- b. Graf yang tidak memiliki sisi disebut graf kosong atau graf nol. Graf nol dengan titik n dan setiap dua titik berbeda dilambangkan dengan N_n . Misalkan graf kosong dengan 3 titik yang disimbolkan dengan N_3



Gambar 2.4 graf kosong dengan 3 titik

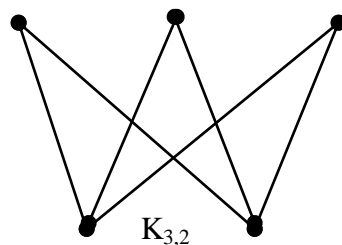
- c. Sebuah graf G disebut bipartisi jika himpunan titik G dapat dipartisi menjadi dua himpunan bagian A dan B sedemikian hingga setiap sisi dari G menghubungkan sebuah titik di A dan sebuah titik di B . Kita sebut (A,B) bipartisi dari G .



Gambar 2.5 Graf G bipartisi

- d. Apabila G sederhana dan bipartisi dengan bipartisi (A,B) sedemikian hingga setiap titik di A berhubungan langsung dengan setiap titik di B , maka G disebut graf bipartisi komplit, dilambangkan dengan $K_{m,n}$ dimana $|A| = m$ dan $|B| = n$.

Sebagai contoh, perhatikan graf pada Gambar 2.6.

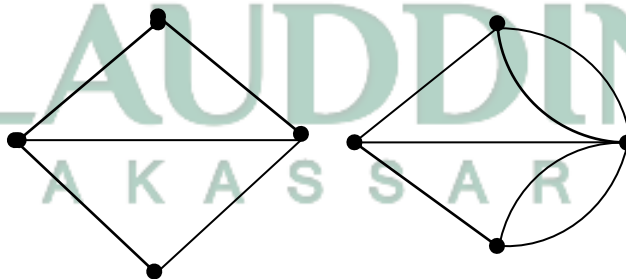


Gambar 2.6 Graf $K_{3,2}$ bipartisi komplit

Perhatikan bahwa, sebagai akibat dari definisi, graf bipartisi mungkin saja mempunyai sisi rangkap, tetapi tidak mungkin memuat gelung. Begitu juga banyaknya titik graf bipartisi komplet $K_{m,n}$ adalah $m+n$ dan banyaknya sisi adalah mn .¹⁰

3. *Graf tak Berarah(Undirected graf)*

Graf yang setiap sisinya tidak mempunyai arah anak panah tetapi memiliki bobot pada setiap sisinya. Urutan pasangan titik yang terhubung oleh sisi tidak diperhatikan. Sehingga $(u,v) = (v,u)$ adalah sisi yang sama. Sehingga graf tak berarah sering dipakai pada jaringan saluran telepon karena sisi pada graf tak berarah menyatakan bahwa saluran telepon dapat beroperasi pada dua arah.



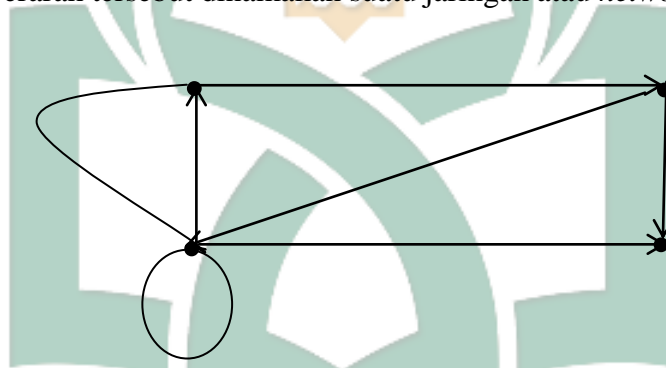
Gambar 2.7 Graf tak berarah¹¹.

¹⁰Wahyuni Abidin, “*Matematika Diskrit*”, (Makassar: Buku Daras UIN Alauddin, 2013), h. 81-85.

¹¹<http://yuliee.wordpress.com/2010/04/22/graf-tak-berarah/> (Online), (9 Juli 2014).

4. Graf Berarah(*directed graf*)

Graf berarah merupakan graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Konsep graf berarah lebih sering digunakan dibandingkan dengan konsep graf tak berarah. Apabila ruas suatu graf berarah mempunyai suatu bobot, graf berarah tersebut dinamakan suatu jaringan atau *network*.



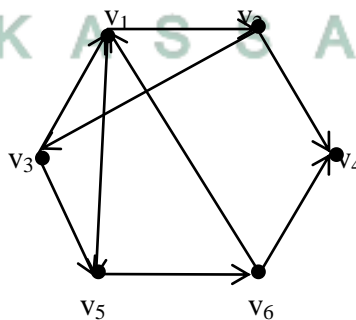
Gambar 2.8 Graf berarah

Beberapa Pengertian dalam graf berarah :

1. Derajat ke luar (out degree) suatu titik adalah banyaknya ruas yang mulai / keluar dari titik tersebut.
2. Derajat ke dalam (in degree) suatu titik adalah banyaknya ruas yang berakhir / masuk ke titik tersebut.
3. Titik berderajat ke dalam = 0 disebut sumber (source), sedangkan titik berderajat ke luar = 0 disebut muara (sink).
4. Pengertian Walk, Trail, Path (Jalur) dan Sirkuit (Cycle) berlaku pula pada graf berarah, dimana harus sesuai dengan arah ruas. Kalau tidak sesuai dengan arah ruas-nya, maka disebut sebagai semi walk, semi path atau semi trail.

5. Path dan Sirkuit(Cycle)

Sebuah Path W dalam sebuah graf berarah G adalah sebuah barisan berganti dari titik-titik dan sisi-sisi berarahnya yang membentuk $W = \{v_0, e_1, e_2, v_2, \dots, e_n, v_n\}$, Sedemikian sehingga setiap sisi e_i mulai pada v_{i-1} dan berakhir di v_i . Jika disini tidak ada arti lain, maka kita nyatakan W dengan barisan titiksnnya atau barisan sisinya. Ingat bahwa sebuah path dalam sebuah graf berarah bisa juga tertutup, dalam hal dimana titik pertama dan terakhir dalam barisan yang sama. Sebuah path sederhana dalam sebuah graf berarah G adalah sebuah path di G dimana setiap titiknya berbeda. Sebuah Sebuah cycle dalam graf berarah adalah sebuah path di G dimana semua titiknya berbeda kecuali yang pertama dan terakhir. Path harus memenuhi syarat yang diperlukan pada arah busurnya. Perhatikan graf G pada gambar 2.9. Tentukan dua path dari v_1 ke v_6 .



Gambar 2.9 Graf G

Ada banyak path dari v_1 ke v_6 . Dua diantaranya adalah (v_1, v_5, v_6) dan $(v_1, v_2, v_3, v_1, v_5, v_6)$. Catatan bahwa barisan dari titik-titik $v_1, v_2, v_4,$

v_6 bukan sebuah path yang dari v_4 ke v_6 karena busur yang menghubungkan v_1 ke v_6 tidak dimula pada v_4 yaitu busur bukan titik dalam arah yang sama seperti path.¹²

6. Graf Berarah Terhubung

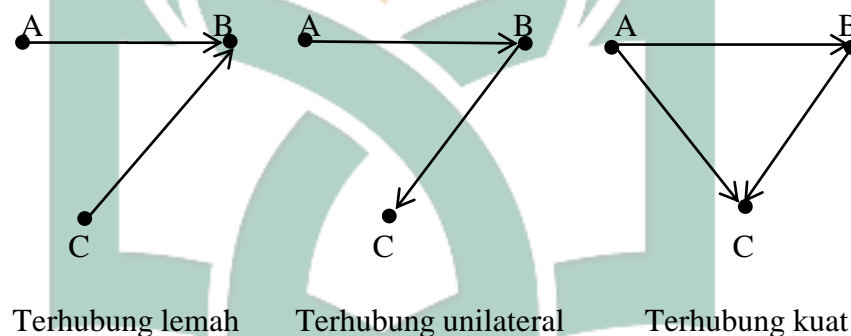
Dua buah titik v_1 dan titik v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2 . G disebut graf terhubung (*connected graf*) jika untuk setiap pasang titik v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j . Jika tidak, maka G disebut graf tak terhubung (*disconnected graf*). Graf berarah G dikatakan terhubung jika graf tidak berarahnya terhubung (graf tidak berarah dari G diperoleh dengan menghilangkan arahnya). Pada graf berarah terdapat 3 pengertian keterhubungan, yakni :

1. Terhubung lemah, jika terdapat suatu semi path antara setiap 2 titik dari D . Jika u dan v tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya, maka u dan v dikatakan terhubung lemah (*weakly connected*).
2. Terhubung unilateral, jika antara setiap 2 titik u dan v dari D , terdapat jalur dari u ke v atau dari v ke u .

¹²Seymour Lipshutz, Mar Lars Lipson, “*Matematika Diskrit*”, (Jakarta: Salemba Teknik, 2002), h. 105-106.

3. Terhubung kuat, jika antara setiap 2 titik u dan v dari D , terdapat jalur dari u ke v dan dari v ke u . Dua titik, u dan v , pada graf berarah G disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari u ke v dan juga lintasan berarah dari v ke u .

Contoh:



Gambar 2.10 Graf terhubung

B. Lintasan Terpendek

Lintasan terpendek adalah lintasan minimum yang diperlukan untuk suatu tempat dari tempat tertentu. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf berbobot, Yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Dalam kasus ini yang dimaksud berupa jarak. Dalam hal ini bobot harus bernilai positif, pada lain hal terdapat bobot dengan nilai negatif. Lintasan terpendek dengan titik awal s dan titik tujuan t didefinisikan sebagai lintasan dari s ke t dengan bobot minimum dan berupa lintasan sederhana(*simple path*).

Salah satu aplikasi graf berarah berlabel yang sering dipakai adalah mencari lintasan terpendek diantara 2 titik. Apabila masalahnya adalah mencari lintasan terpendek tetap dapat digunakan dengan cara mengganti nilai sisi.

Definisi 2.9. Misalkan G adalah suatu graf, untuk v dan w adalah titik dalam G . suatu Walk dari v ke w adalah barisan titik dan sisi secara berselang-seling, diawali dari titik v dan diakhiri pada titik w . Walk dengan panjang n dari v ke w ditulis : $v_0 e_1 v_1 e_2 v_2 \dots v_{n-2} e_n v_n$ dengan $v_0 = v$; $v_n = w$; v_{i-1} dan v_i adalah titik-titik ujung sisi e_i . Lintasan dengan panjang n dari v ke w adalah walk dari v ke w yang semua sisinya berbeda. Lintasan dari v ke w dituliskan sebagai $v = v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n = w$ dengan $e_i \neq e_j$ untuk $i \neq j$. Penulisan berikutnya akan dipergunakan notasi

$$v_1 \rightarrow v_1, A = \{v_1 \rightarrow v_2, v_2 \rightarrow v_3, \dots\}$$

Definisi 2.10. Lintasan tertutup adalah suatu barisan sisi $(e_{i1}, e_{i2}, \dots, e_{in})$ sedemikian rupa sehingga titik terminal e_{ij} berimpit dengan titik awal $e_{i(j+1)}$ untuk $1 \leq j \leq k - 1$.

Pada Gambar 2.2 terdapat:

- Pada titik $v_1 \rightarrow e_1 \rightarrow v_2 \rightarrow e_3 \rightarrow v_3 \rightarrow e_4 \rightarrow v_3 \rightarrow e_5 \rightarrow v_4$ semua sisi berbeda (e_1, e_3, e_4 , dan e_5) masing-masing muncul sekali. Ada titik yang berulang (v_3 muncul 2 kali). Titik awal dan titik akhir tidak sama dengan titik awal = v_1 dan titik akhir = v_4 , Barisan ini merupakan lintasan dari $v_1 \rightarrow v_4$ dengan panjang 4.

- b. Pada titik $v_1 \rightarrow e_1 \rightarrow v_2 \rightarrow e_3 \rightarrow v_3 \rightarrow e_5 \rightarrow v_4 \rightarrow e_5 \rightarrow v_3 \rightarrow e_6 \rightarrow v_5$ ada sisi yang muncul lebih dari sekali, yaitu e_5 (muncul 2 kali) berarti barisan tersebut merupakan walk dari $v_1 \rightarrow v_5$ dengan panjang lima.

Terdapat beberapa macam persoalan lintasan terpendek antara lain :

- Lintasan terpendek antara dua buah titik tertentu (a pair shortest path)
- Lintasan terpendek antara semua pasangan titik (all pairs shortest path)
- Lintasan terpendek dari titik tertentu ke semua titik yang lain (single-source shortest path).
- Lintasan terpendek antara dua buah titik yang melalui beberapa titik tertentu (intermediate short path).¹³

C. Algoritma Bellman-Ford

Algoritma Bellman-ford adalah algoritma untuk menyelesaikan permasalahan lintasan terpendek dengan dengan sumber tunggal. Jadi algoritma Bellman-ford menghitung jarak terpendek (dari satu sumber) pada sebuah graf berbobot. Maksud dari sumber tunggal ialah bahwa algoritma menghitung semua jarak terpendek yang berawal dari satu titik. Di samping itu algoritma ini menggunakan $d[u]$ sebagai batas atas dengan jarak $d[u,v]$ dari u ke v . Algoritma ini melakukan inisialisasi jarak titik sumber ke titik nol dan semua titik lainnya (sampai tak hingga). Secara

¹³Munir, Rinaldi, "Matematika Diskrit Program Studi Informatika Diktat kuliah IF2091", Institute Teknologi bandung, <http://id.wikipedia.org>. Makalah IF2091, 2009.

progresif algoritma ini melakukan perbaikan (updating) jarak pada setiap titik sumber ke titik v di dalam V hingga dicapai lintasan dalil Boolean *TRUE* yaitu jika grafik mengandung lingkaran tidak negatif maka titik dapat dicapai dari titik sumber, dan dalam kondisi lain dikatakan Boolean *FALSE*. Algoritma Bellman ford sebagai berikut¹⁴:

BELLMAN-FORD (G, w, s)

1. INITIALIZE-SINGLE-SOURCE (G, s)
2. for each vertex $i = 1$ to $V[G] - 1$ do
3. for each edge (u, v) in $E[G]$ do
4. RELAX (u, v, w)
5. For each edge (u, v) in $E[G]$ do
6. if $d[u] + w(u, v) < d[v]$ then
7. return FALSE
8. return TRUE

a. Langkah-langkah algoritma Bellman-Ford

Algoritma Bellman-Ford menentukan lintasan terpendek

Secara umum, langkah-langkah algoritmanya adalah sebagai berikut:

- a. Menentukan titik asal dan mendaftar semua titik maupun sisi
- b. Memberi nilai untuk titik asal sama dengan nol dan titik-titik lainnya dengan tak hingga.

¹⁴Eko Budi Purwanto, “Perancangan dan Analisis Algoritma”, (Yogyakarta: Penerbit Graha Ilmu, 2008), h.129-130.

- c. Memulai iterasi terhadap semua titik yang ada dimulai dengan titik asal. untuk menentukan jarak dari semua titik yang berhubungan dengan titik asal dengan formula seperti berikut:

U = titik asal

V = titik tujuan

UV = sisi yang menghubungkan U dan V,

Jika jarak V lebih besar dari jarak U+bobot UV maka jarak V diisi dengan jarak U+bobot UV, dilakukan hingga semua titik terjelajahi.

- d. Melakukan iterasi untuk semua sisi yang ada untuk mengecek apakah ada siklus negatif dalam graf tersebut, kemudian melakukan pengecekan seperti dibawah ini:

Untuk semua sisi UV, jika jarak V lebih besar dari jarak U+bobot UV maka sudah jelas bahwa graf tersebut memiliki siklus negatif.

b. Kompleksitas Waktu algoritma Bellman-Ford

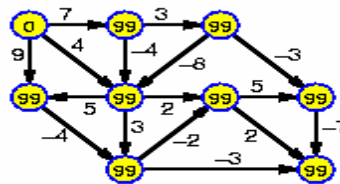
Secara kompleksitas waktu, algoritma ini hanya memiliki 1 kemungkinan, yaitu dengan notasi *Big-O* diberikan $O(V \cdot E)$. V adalah jumlah titik graf berbobot, lalu E adalah jumlah sisi dalam graf. Oleh sebab itu untuk jumlah sisi ataupun titik yang sangat besar akan menyebabkan algoritma ini akan berjalan lebih lama dibandingkan algoritma Dijkstra.

Berikut penjabaran perhitungan kompleksitas waktu.

1. Tahap inisialisasi mempunyai kompleksitas $O(V)$. Maksudnya bahwa, jika n banyaknya titik maka inisialisasi dilakukan sebanyak n dengan titik asal 0 dilakukan satu kali dan titik yang lainnya ($n-1$).
2. Tahap kedua yaitu melakukan pencarian jalur atau jalan terpendek terhadap suatu titik s mempunyai kompleksitas $O(V.E)$. Hal ini karena perulangan untuk tahap pencarian ada dua kali, perulangan pertama sebanyak $O(E)$, selanjutnya $O(E)$ diulanga sebanyak $O(V)$ jadi totalnya adalah $O(V.E)$
3. Tahap ketiga yaitu pengecekan ada atau tidaknya sisi negative pada jalur, mempunyai kompleksitas $O(E)$.

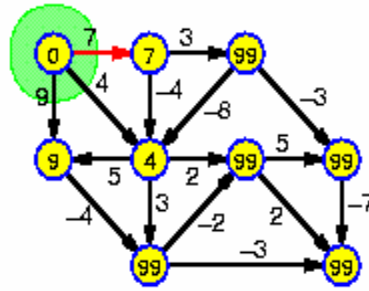
Dari semua kompleksitas di atas dapat ditentukan kompleksitas waktu algoritma ini adalah $O(V.E)$.¹⁵

Skema Pencarian Lintasan Terpendek dengan algoritma Bellman Ford



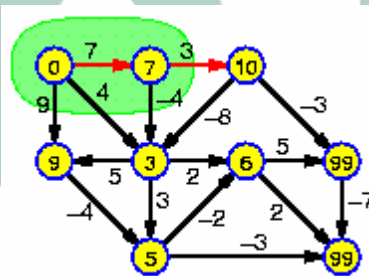
Gambar 2.11 Graf berbobot negatif.

¹⁵Andana, Galih, "Algoritma Bellman – Ford dalam distance Vektor Routing Protocol", <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2009-2010/Makalah2009/MakalahIF3051-2009-044.pdf>, 2009, (5 Mei 2014)



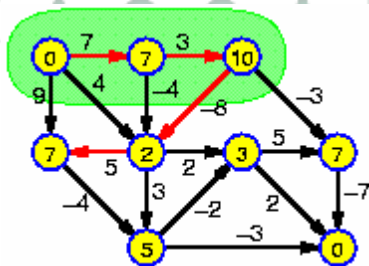
Gambar 2.12 Tahap pertama Algoritma

Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11.



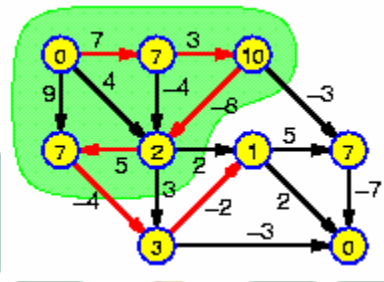
Gambar 2.13 Tahap Kedua Algoritma

Gambar 2.13 Tahap kedua Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11



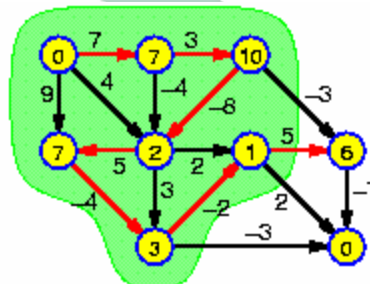
Gambar 2.14 Tahap Ketiga Algoritma

Gambar 2.14 Tahap ketiga Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11.



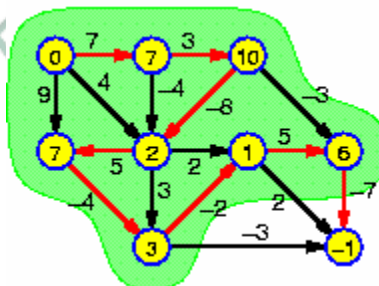
Gambar 2.15 Tahap Keempat Algoritma

Gambar 2.15 Tahap keempat Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11.



Gambar 2.16 Tahap Kelima Algoritma

Gambar 2.16 Tahap Kelima Algoritma Bellman-Ford untuk penyelesaian contoh graf pada gambar 2.11.



Gambar 2.17 Tahap Keenam Algoritma

Gambar 2.17 Lintasan terpendek untuk penyelesaian contoh graf pada gambar 2.11 sebesar -1.¹⁶

¹⁶Kamayudi , Apri, “ *Studi Dan Implementasi Algoritma, Dijkstra, Bellman - Ford Dan Floyd–Warshall Dalam Menangani Masalah Lintasan Terpendek Dalam Graf*”,

D. Matlab

1. Sekilas Matlab

Matlab singkatan dari *Matrix Laboratory*. Matlab merupakan bahasa pemrograman yang dikembangkan oleh Mathwork. Inc. Pada awalnya, program ini merupakan interface untuk koleksi rutin-rutin numerik dari proyek *LINPACK* dan *EISPACK*, namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc. Matlab telah berkembang menjadi sebuah lingkungan pemrograman yang canggih yang berisi fungsi-fungsi built-in untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulus matematis lainnya. Matlab juga berisi toolbox yang berisi fungsi-fungsi tambahan untuk aplikasi khusus.¹⁷

Matlab memberikan sistem interaktif yang menggunakan konsep array/matrik sebagai standar variabel elemennya tanpa membutuhkan pendeklarasian array seperti pada bahasa lainnya. Kehadiran Matlab memberikan jawaban sekaligus tantangan. Matlab menyediakan beberapa pilihan untuk dipelajari, mempelajari metoda visualisasi saja, pemrograman saja atau kedua-duanya. Kemudahan yang ditawarkan sama sekali bukan tandingan bahasa pemrograman yang lain, karena bahasa pemrograman yang lain memang tidak menawarkan kemudahan serupa. Matlab memang dihadirkan bagi orang-orang yang tidak ingin disibukkan dengan rumitnya sintak dan alur logika pemrograman, sementara pada saat

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2006-2007/Makalah/Makalah0607-115.pdf>, 2007, (4 Mei 2014).

¹⁷ Dedy Barnabas Lasfeto, Oky Dwi Nurhayati, “*Analisis Statistika Deskriptif menggunakan Matlab*”, (Yogyakarta: Graha Ilmu, 2008), h.22.

yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung pekerjaannya. Selain itu Matlab juga memberikan keuntungan bagi programmer-developer program yaitu untuk menjadikan program perbandingan yang sangat handal, hal tersebut dapat dilakukan karena kemampuannya akan fungsi matematika, fisika, statistik dan visualisasi.¹⁸

Kegunaan Matlab secara umum adalah:

- a. Matematika dan Komputasi
- b. Pengembangan dan algoritma
- c. Pemodelan, simulasi dan pembuatan prototype
- d. Analisis data, eksplorasi dan visualisasi
- e. Pembuatan aplikasi termasuk pembuatan grafical user interface.¹⁹

2. *Lingkungan Kerja Matlab*

Sebagaimana bahasa pemrograman lainnya, Matlab juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam pembangunan aplikasi. Pada setiap versi Matlab yang terbaru, lingkungan terpadunya akan semakin dilengkapi. Lingkungan terpadu ini terdiri atas beberapa form/window yang memiliki kegunaan masing-masing. Untuk memulai aplikasi Matlab, anda hanya perlu mengklik ikon Matlab pada desktop window, atau bisa juga dengan menu start seperti pada aplikasi-aplikasi lainnya. Setiap pertama kali mulai membuka aplikasi Matlab, anda

¹⁸Gunaidi Abdia Away, "The Shortcut of Matlab Programming", (Bandung: Informatika, 2006), h. 2.

¹⁹Dedy Barnabas Lasfeto, Oky Dwi Nurhayati, "Analisis Statistika Deskriptif menggunakan Matlab", :Yogyakarta: Graha Ilmu, 2008), h. 23.

akan memperoleh beberapa form/window, yang sebenarnya menurut penulis hanya membuat desktop anda kelihatan penuh. Anda dapat menutup semua window tersebut kecuali command window yang menjadi window utama Matlab. Matlab akan menyimpan mode/setting terakhir lingkungan Kerja yang anda gunakan sebagai mode/setting lingkungan kerja pada saat anda membuka aplikasi Matlab di waktu berikutnya.²⁰



²⁰Gunaidi Abdia Away, *"The Shortcut of Matlab Programming"*, (Bandung: Informatika, 2006), h.5.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian ini yang digunakan adalah studi kasus yang bertujuan mengumpulkan informasi yaitu nama-nama jalan yang ada di kecamatan Biringkanaya, kemudian menentukan jarak antara semua batas-batas jalan dengan menggunakan aplikasi google Maps. Kemudian diselesaikan dengan menggunakan algoritma Bellman-Ford.

B. Waktu dan Lokasi Penelitian

Waktu yang digunakan dalam pelaksanaan penelitian ini adalah sekitar 2 bulan terhitung dari Januari sampai dengan Februari 2015 dan lokasi penelitian pada Perpustakaan Umum UIN Alauddin Makassar.

C. Prosedur Penelitian

Untuk menjawab permasalahan yang ada, digunakan prosedur penelitian dengan langkah-langkah sebagai berikut:

1. Melakukan pengumpulan data

Pengumpulan data diperoleh dari internet menggunakan aplikasi google Maps yang dilakukan di Perpustakaan Umum UIN Alauddin Makassar

2. Mencari jalur minimum Tukang Bentor menggunakan algoritma Bellman-Ford secara manual. Adapun langkah-langkahnya:

- a. Menentukan titik asal dan mendaftar semua titik maupun sisi

b. Memberi nilai untuk titik asal sama dengan nol dan titik-titik lainnya dengan tak hingga.

c. Memulai iterasi terhadap semua titik yang ada dimulai dengan titik asal, untuk menentukan jarak dari semua titik yang berhubungan dengan titik asal dengan formula seperti berikut:

U = titik asal

V = titik tujuan

UV = sisi yang menghubungkan U dan V,

Jika jarak V lebih besar dari jarak U+bobot UV maka jarak V diisi dengan jarak U+bobot UV, dilakukan hingga semua titik terjelajahi.

3. Mencari jalur minimum Tukang Bentor menggunakan algoritma Bellman-Ford dengan bantuan program Matlab.

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R

BAB IV

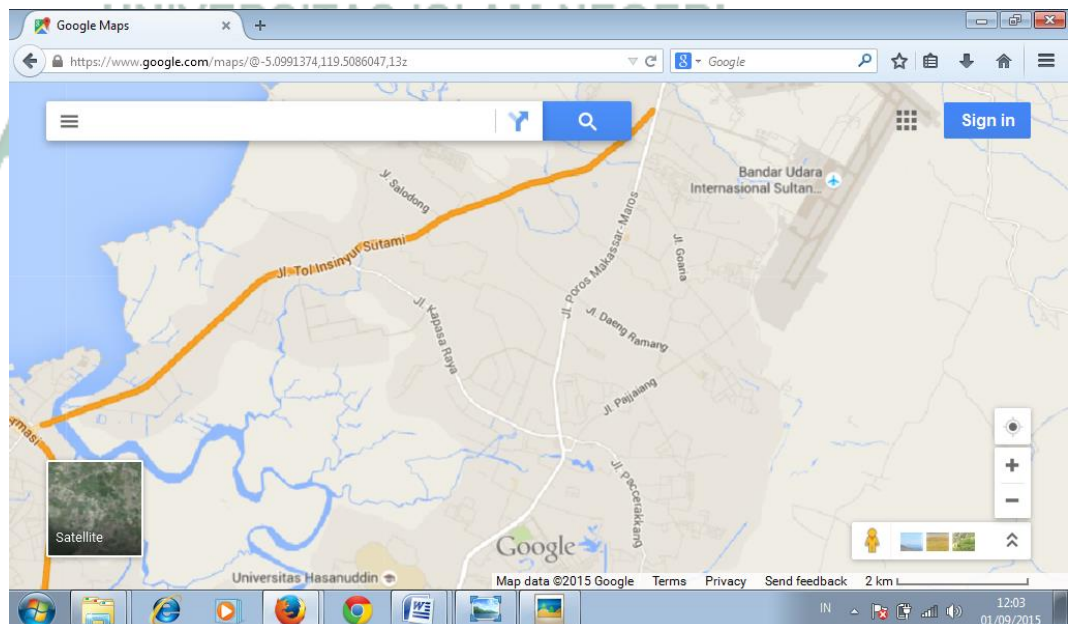
HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Berdasarkan rumusan masalah dan prosedur penelitian, dilakukan beberapa tahapan untuk mencapai tujuan penelitian. Tahapan tersebut adalah sebagai berikut.

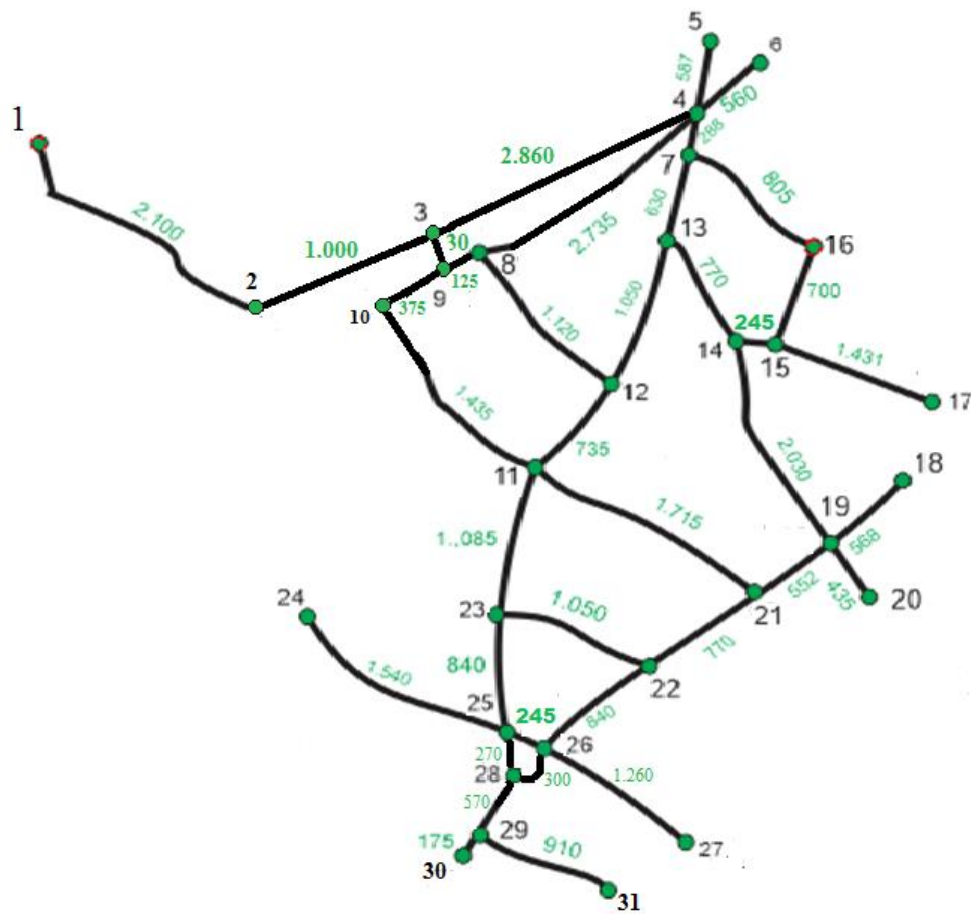
1. Pengumpulan Data

Berdasarkan penelitian yang dilakukan dengan menggunakan aplikasi google Maps diperoleh peta kecamatan Biringkanaya. Berikut peta kecamatan Biringkanaya yang dijadikan sampel dalam penelitian ini yaitu:



Gambar 4.1. Peta google Maps Kecamatan Biringkanaya

Pada peta kecamatan Biringkanaya ada 15 jalan yang akan diteliti yang dituangkan dalam bentuk graf. Adapun gambarnya sebagai berikut:



Gambar 4.2 Graf rute jalan kecamatan Biringkanaya

Keterangan:

1 - 2 : Jl. Salodong

2 - 3 - 4 - 6 : Jl. Ir. Sutami (Sebelah kiri Tol)

3 - 9 : Terowongan

5 – 4 – 7 – 13 – 12 – 11 – 23 – 25 – 28 – 29 – 30 : Jl. Perintis

Kemerdekaan

7 – 8 – 9 10 : Jl. Ir. Sutami (Sebelah kanan tol)

7 – 16 – 15 : Jl. Arung Teko

8 – 12 : Jl. H. Abd. Jabbar

10 – 11: Jl Batara Bira

13 – 14 – 19 – 20 : Jl. Gowa ria

14 – 15 – 17 : Jl. Bakung

11 – 21 : Jl. Dg. Rammang

19 – 18 Jl. Laikang

21 – 22 – 26 - 28 : Jl. Pajaiang B

23 – 22 : Jl. Sanrangang

24 – 25 : Kapasa Raya

25 – 26 – 27 : Jl. Paccerakang

29 – 31 : Jl. Lanraki

Pada Gambar graf di atas terdapat 31 titik. Titik-titik tersebut merupakan batas-batas jalan, dimana titik 1 merupakan titik awal (pangkalan Bentor). Di kecamatan Biringkanaya ini terdapat jalur satu arah yang terdapat di daerah Jalan Tol yaitu titik 3→4, titik 7→ 8→9→10 dan di daerah Paccerakang yaitu titik 23→24. Data jalan yang digunakan untuk mengetahui jalur terpendek dari pangkalan Bentor ke semua titik (batas jalan). Satuan yang digunakan dalam jarak jalur ini adalah meter.

2. Mencari Rute minimum tukang bentor kecamatan biringkanaya menggunakan Algoritma Bellman_ford

Algoritma Bellman-Ford menghitung jarak terpendek dari satu sumber pada graf berbobot, maksudnya adalah algoritma Bellman-Ford menghitung semua jarak terpendek dari satu titik. Adapun langkah-langkahnya berdasarkan prosedur penelitian sebagai berikut:

- a. Menentukan titik awal dan mendaftar semua titik maupun sisi
- b. Memberi nilai untuk titik awal sama dengan nol dan titik lainnya dengan nilai tak hingga.
- c. Memulai iterasi terhadap semua titik yang dimulai dengan titik asal, untuk menentukan jarak dari semua titik yang berhubungan dengan titik asal dengan cara:

Jika jarak V lebih besar dari jarak $U + \text{bobot } UV$ maka jarak V diisi dengan jarak $U + \text{bobot } UV$.

dimana

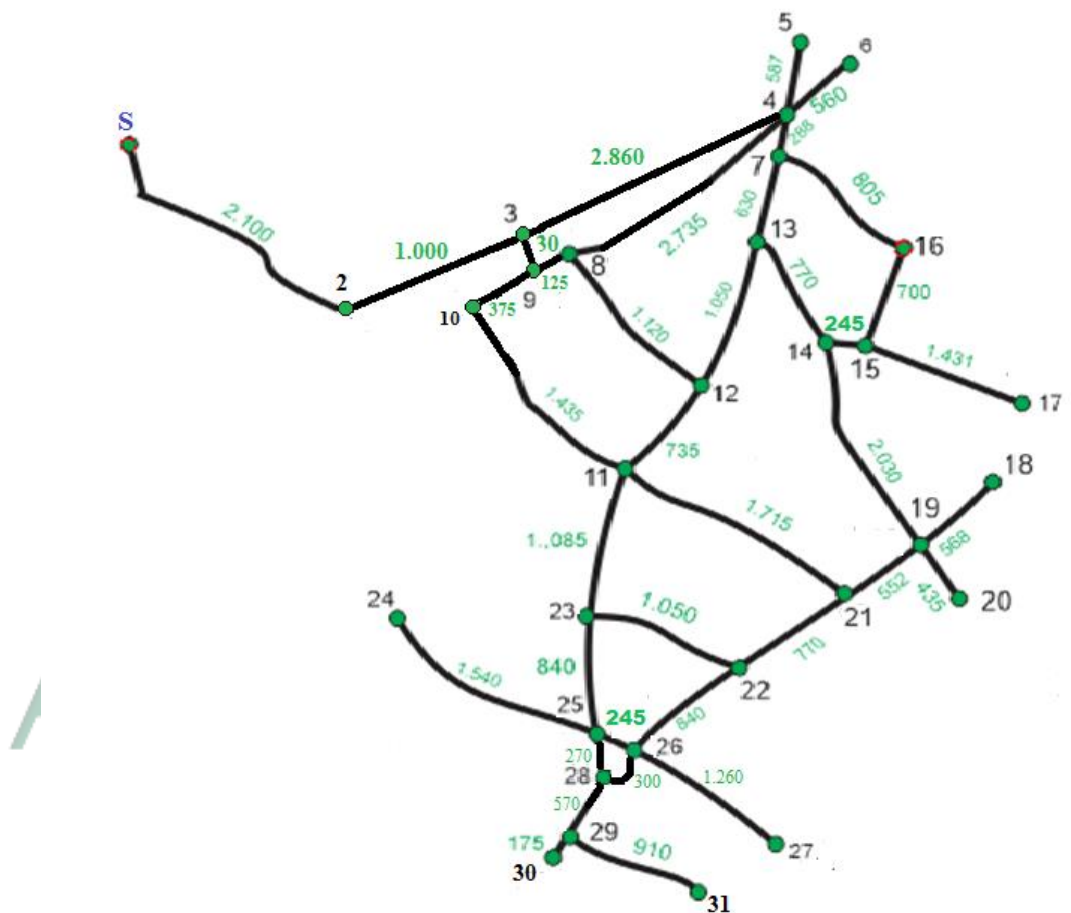
U = titik asal

V = titik tujuan

UV = sisi yang menghubungkan U dan V . Langkah ini dilakukan sehingga semua titik terkunjungi.

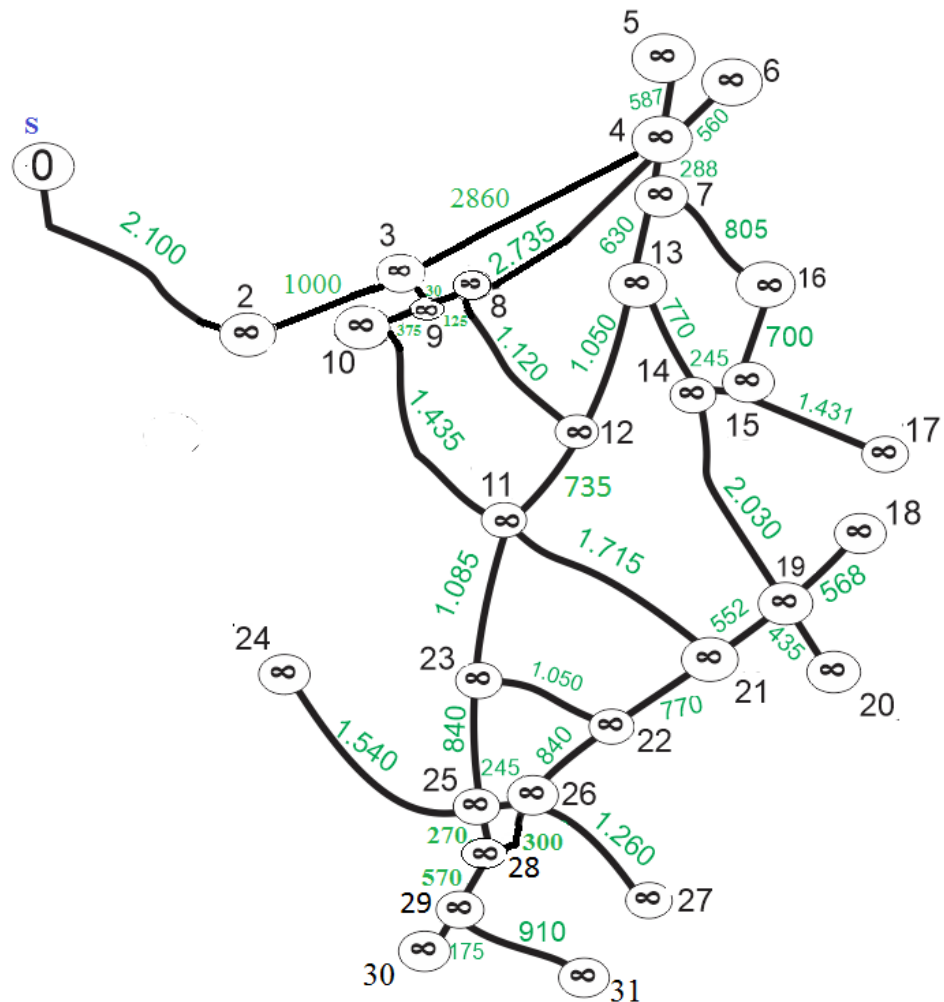
Berikut penyelesaiannya secara manual

Tahap pertama : Menentukan titik 1 sebagai titik awal dan mendaftar semua titik maupun sisi



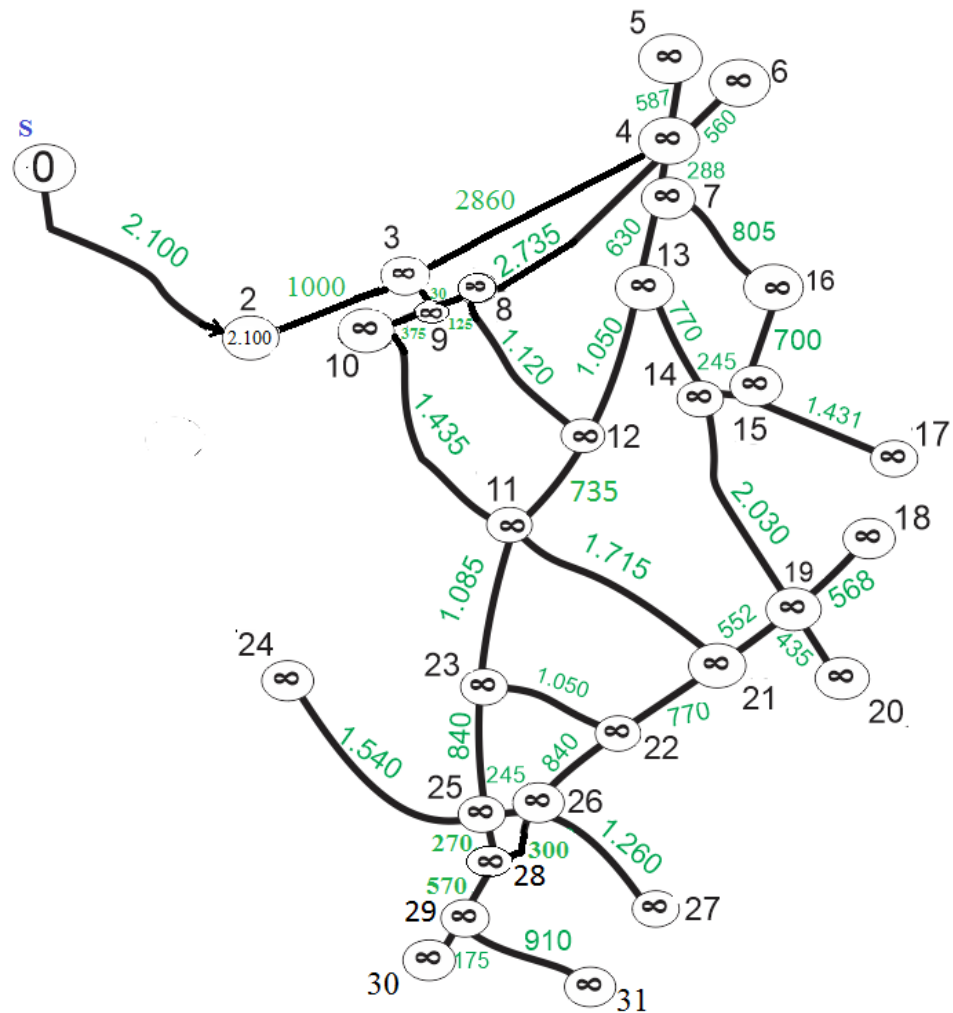
Gambar 4.3 Tahap pertama algoritma bellman-Ford

Tahap kedua : memberi nilai untuk titik awal sama dengan nol dan yang lainnya tak terhingga



Gambar 4.4 Tahap kedua Algoritma Bellman-Ford

Tahap ketiga: iterasi pertama

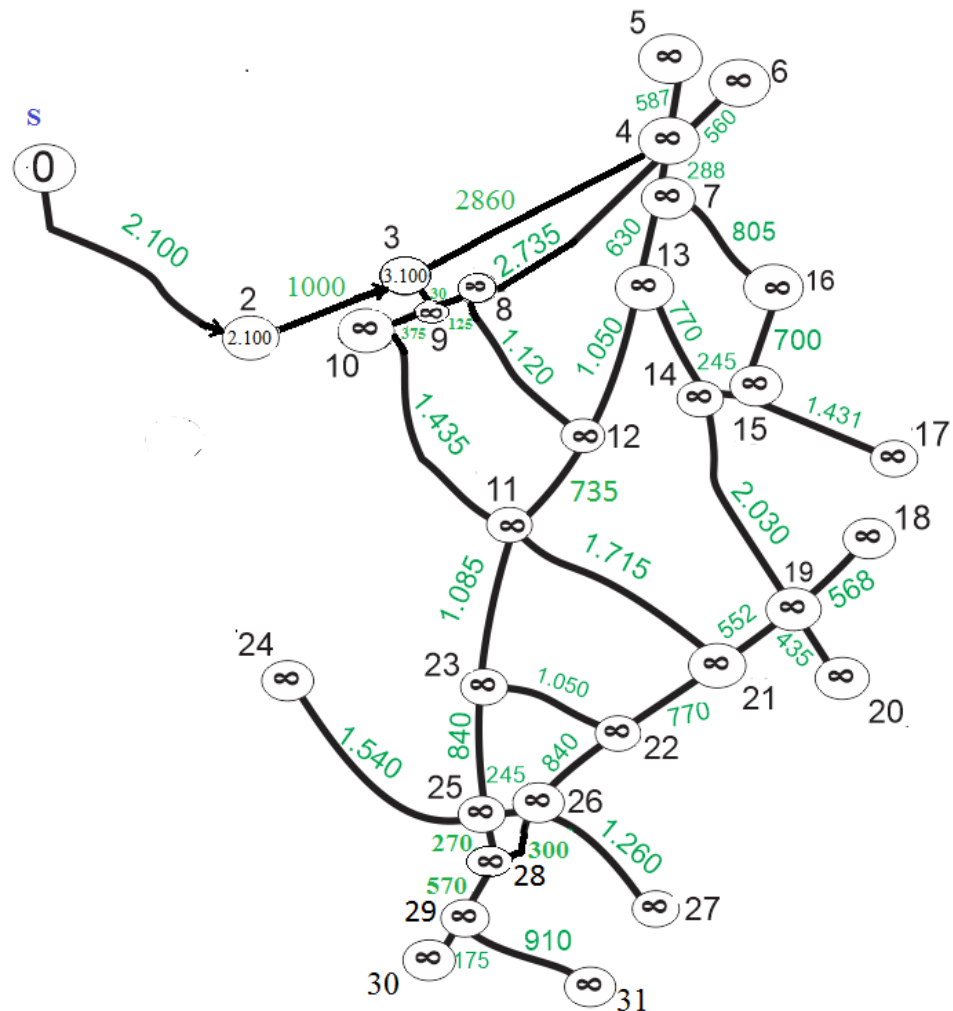


Gambar 4.5 Tahap ketiga Algoritma Bellman ford(iterasi pertama)

Iterasi 1

$$1 \rightarrow 2 = 0 + 2100$$

Tahap ketiga : Iterasi kedua



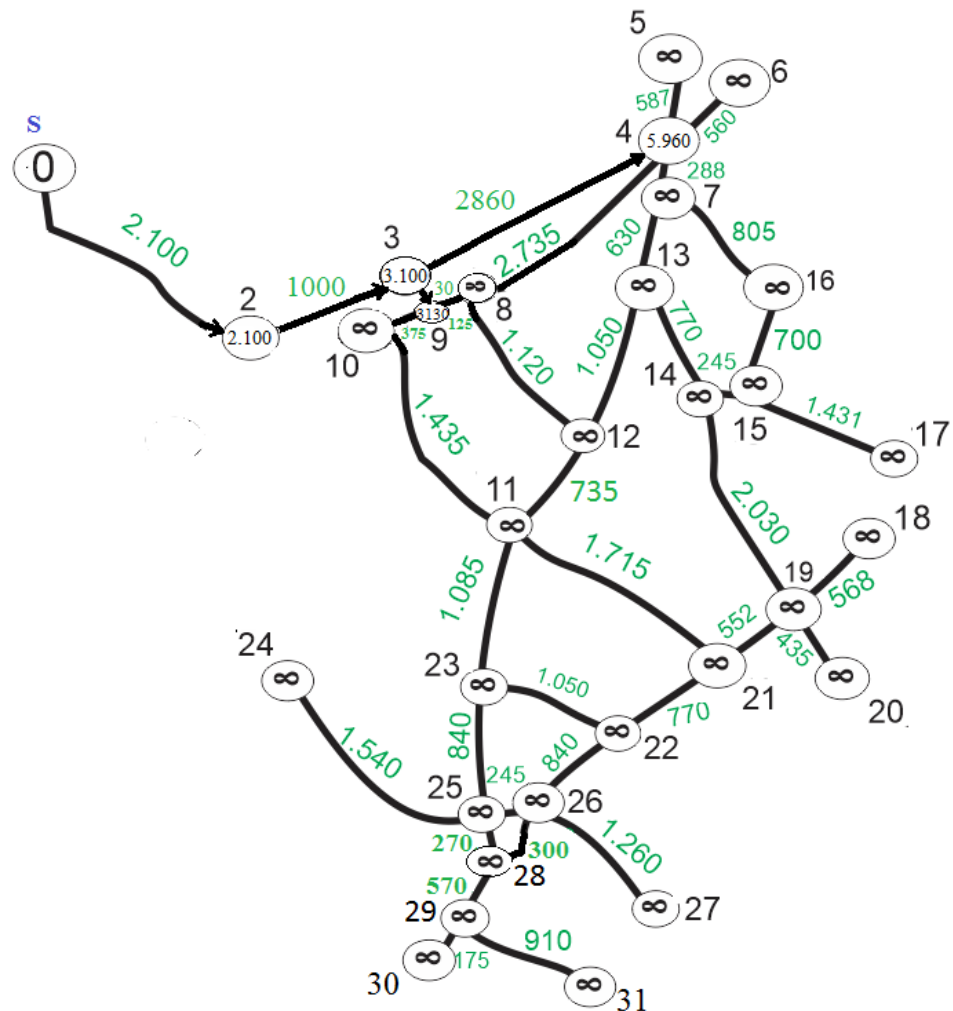
Gambar 4.6 Tahap ketiga algoritma Bellman-Ford(iterasi kedua)

Iterasi 2

$$2 \rightarrow 3 = 2.100 + 1.000$$

$$= 3.100$$

Tahap ketiga : iterasi ketiga



Gambar 4.7 Tahap ketiga algoritma Bellman-Ford(iterasi ketiga)

Iterasi 3

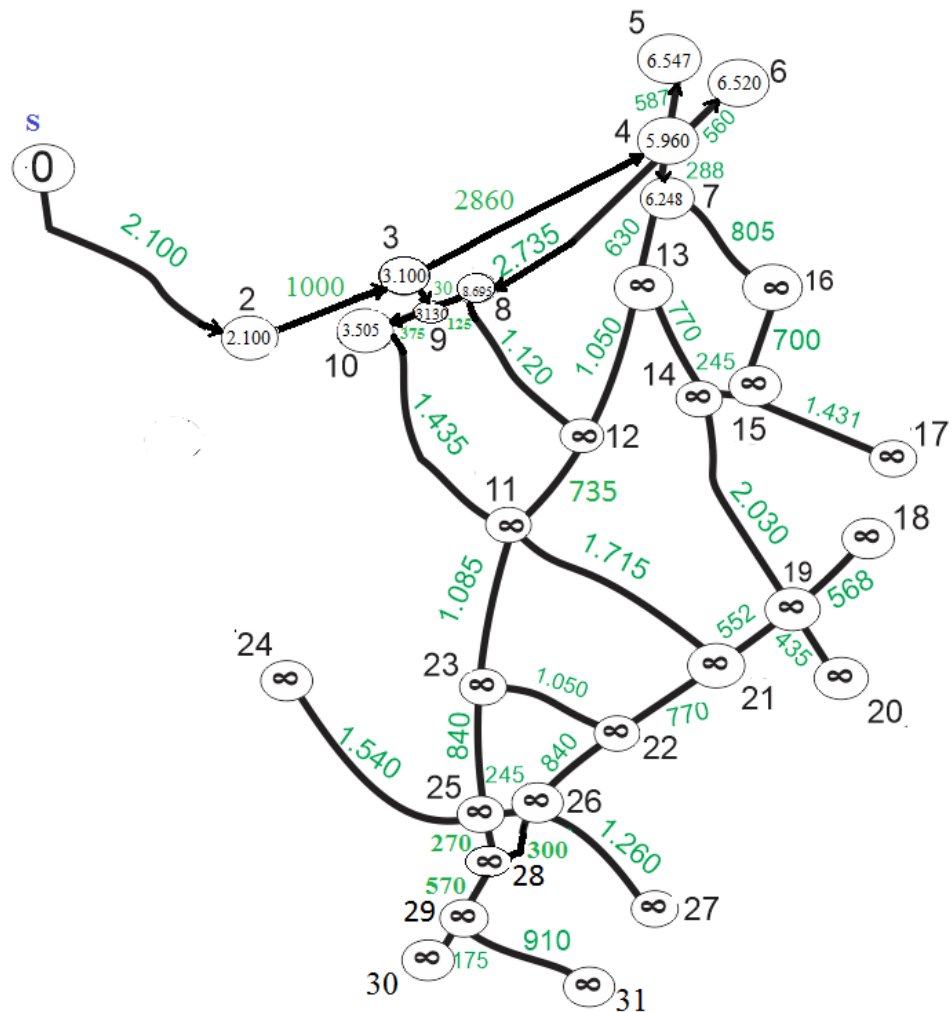
$$3 \rightarrow 4 = 3.100 + 2.860$$

$$= 5.960$$

$$3 \rightarrow 9 = 3.100 + 30$$

$$= 3.130$$

Tahap ketiga : iterasi keempat



Gambar 4.8 Tahap ketiga Algoritma Belman-Ford (Iterasi keempat)

Iterasi 4

$$\begin{aligned} 4 \rightarrow 5 &= 5.960 + 587 \\ &= 6.547 \end{aligned}$$

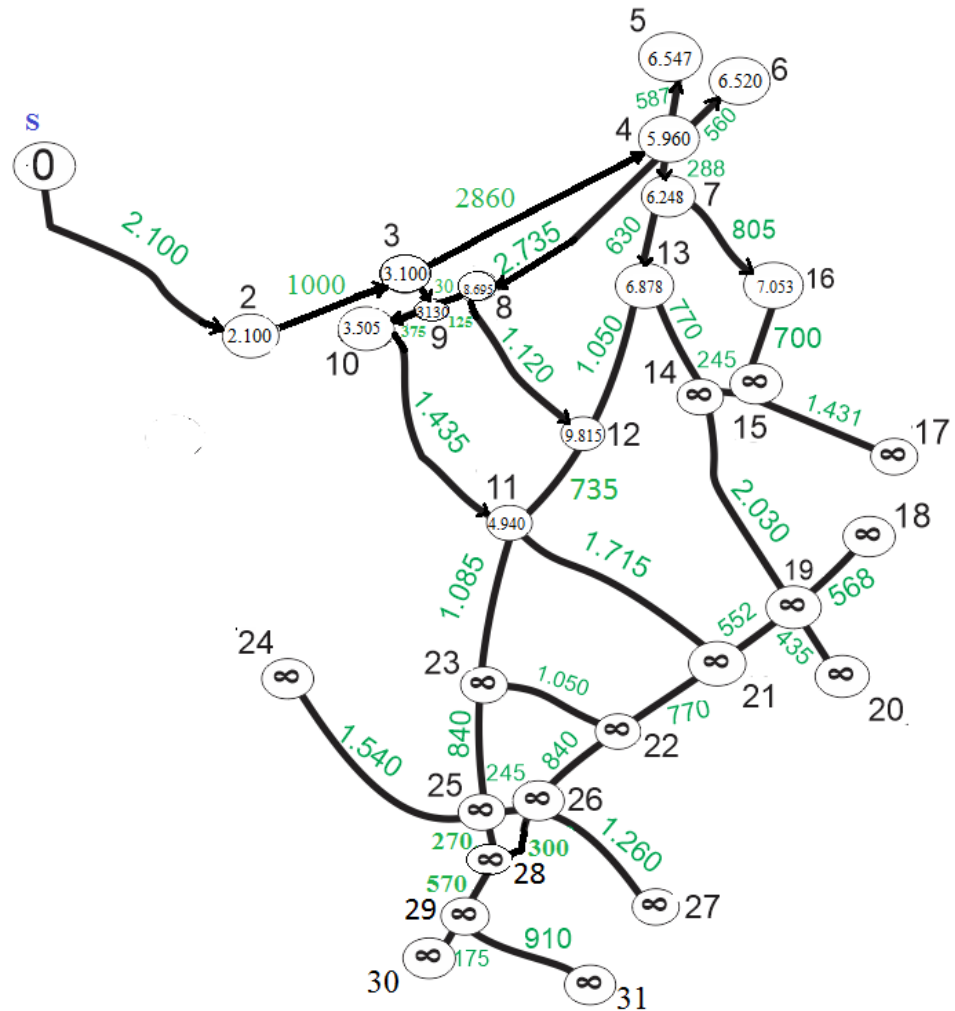
$$\begin{aligned} 4 \rightarrow 8 &= 5.960 + 2.735 \\ &= 8.695 \end{aligned}$$

$$\begin{aligned} 4 \rightarrow 6 &= 5.960 + 560 \\ &= 6.520 \end{aligned}$$

$$\begin{aligned} 9 \rightarrow 10 &= 3.130 + 375 \\ &= 3.505 \end{aligned}$$

$$\begin{aligned} 4 \rightarrow 7 &= 5.960 + 288 \\ &= 6.248 \end{aligned}$$

Tahap ketiga : Iterasi kelima



Gambar 4.9 Tahap ketiga Algoritma Belman-Ford(Iterasi kelima)

Iterasi 5

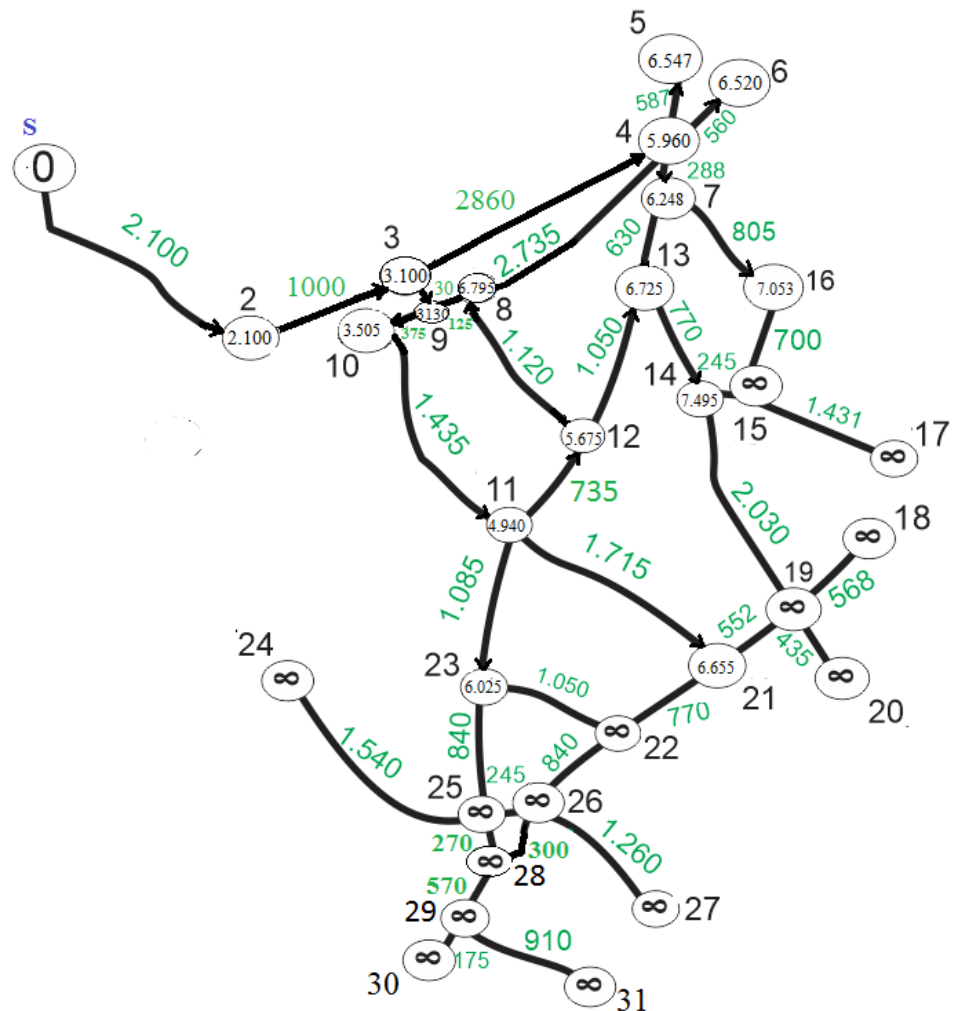
$$\begin{aligned} 7 \rightarrow 13 &= 6.248 + 630 \\ &= 6.878 \end{aligned}$$

$$\begin{aligned} 8 \rightarrow 12 &= 8.695 + 1.120 \\ &= 9.815 \end{aligned}$$

$$\begin{aligned} 7 \rightarrow 16 &= 6.248 + 805 \\ &= 7.053 \end{aligned}$$

$$\begin{aligned} 10 \rightarrow 11 &= 3.505 + 1.435 \\ &= 4.940 \end{aligned}$$

Tahap ketiga :Iterasi keenam



Gambar 4.10 Tahap ketiga Algoritma Belman-Ford(Iterasi keenam)

Iterasi 6

$$11 \rightarrow 12 = 4.940 + 735$$

$$= 5.675$$

$$11 \rightarrow 21 = 4.940 + 1.715$$

$$= 6.655$$

$$11 \rightarrow 23 = 4.940 + 1.085$$

$$= 6.025$$

$$12 \rightarrow 8 = 5.675 + 1.120$$

$$= 6.795$$

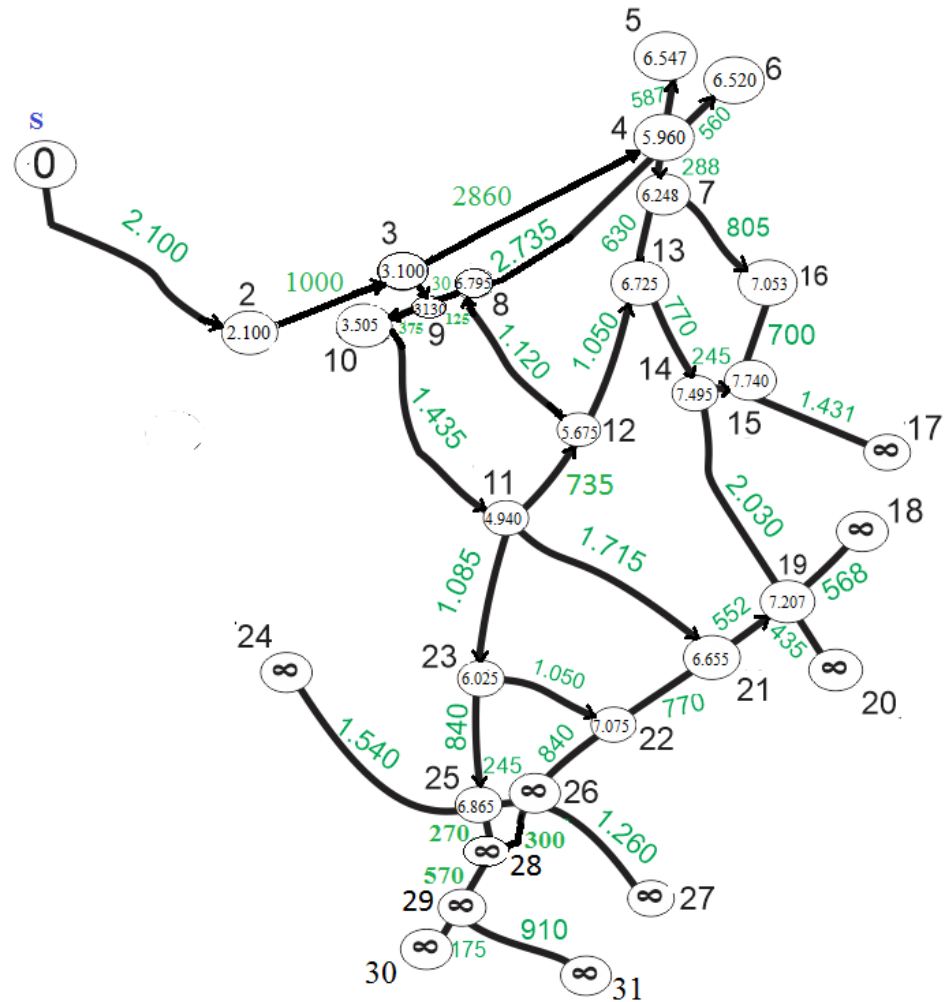
$$12 \rightarrow 13 = 5.675 + 1.050$$

$$= 6.725$$

$$13 \rightarrow 14 = 6.725 + 770$$

$$= 7.495$$

Tahap ketiga : iterasi ketujuh



Gambar 4.11 Tahap ketiga Algoritma Bellman-Ford (iterasi ketujuh)

Iterasi 7

$$14 \rightarrow 15 = 7.495 + 245$$

$$= 7.740$$

$$21 \rightarrow 19 = 6.655 + 552$$

$$= 7.207$$

$$23 \rightarrow 22 = 6.025 + 1.050$$

$$= 7.075$$

$$23 \rightarrow 25 = 6.025 + 840$$

$$= 6.865$$

1

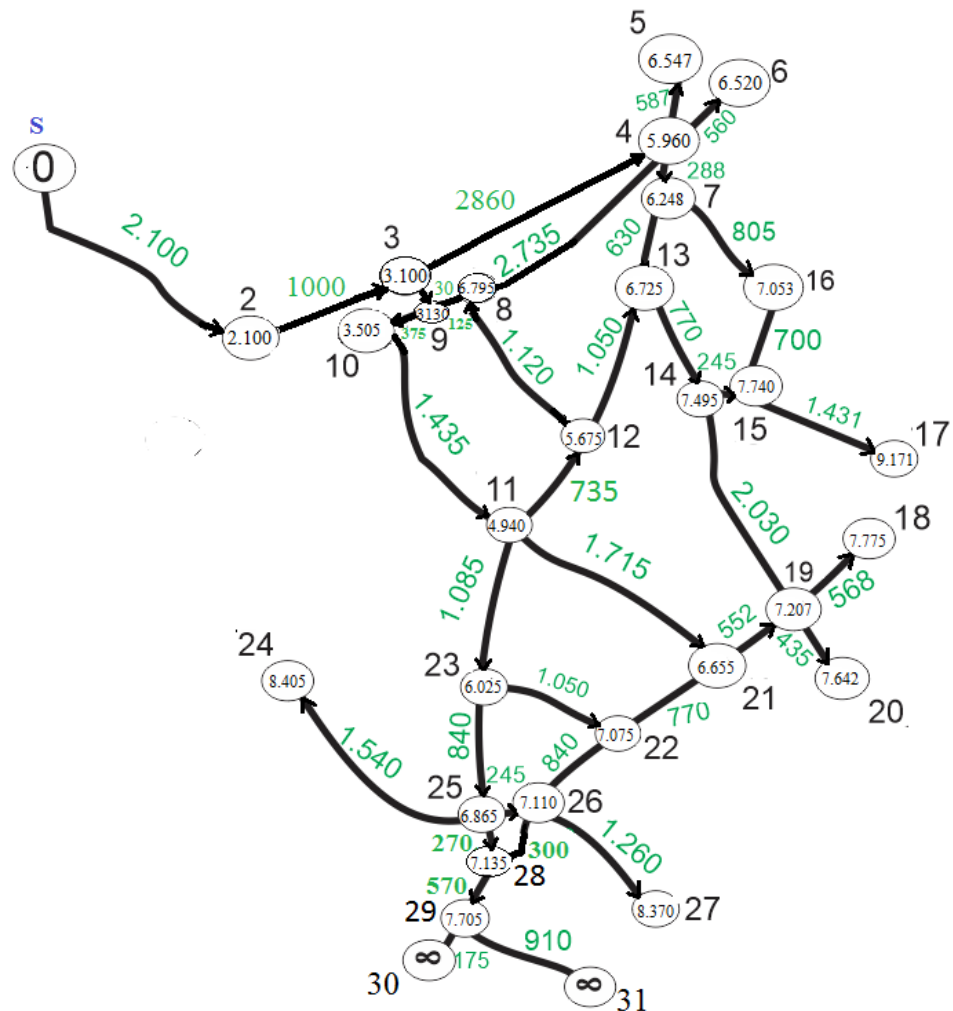


r 4.12 Tahap ketiga Algoritma bellman-Ford(iterasi

$$25 \rightarrow 26 = 6.865 + 245$$
$$= 7.110$$

$$25 \rightarrow 24 = 6.865 + 1.540$$
$$= 8.405$$

$$25 \rightarrow 28 = 6.856 + 270$$
$$= 7.135$$



IVI A A A S S A A

Iterasi 9

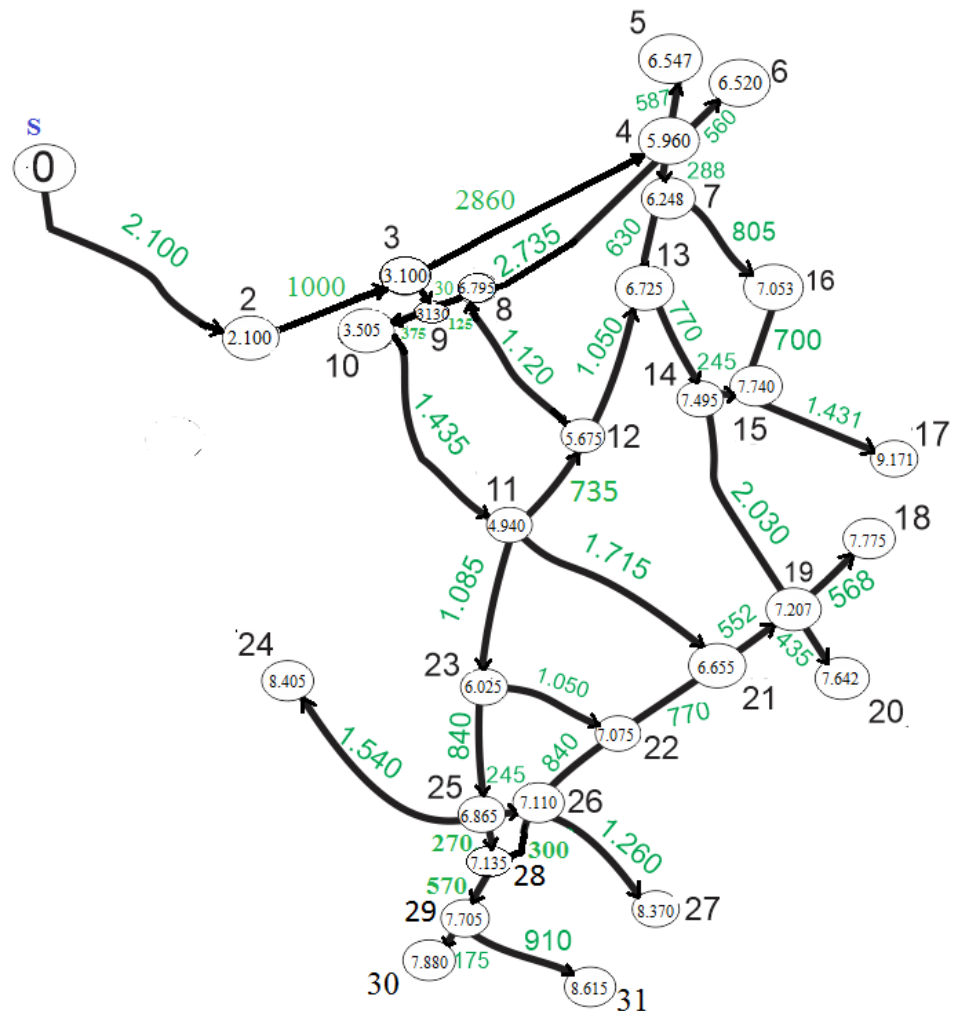
$$26 \rightarrow 27 = 7.110 + 1.260$$

$$= 8.370$$

$$28 \rightarrow 29 = 7.135 + 570$$

$$= 7.705$$

Tahap ketiga : iterasi kesepuluh



Gambar 4.14 Tahap ketiga Algoritma Bellman Ford (Iterasi sepuluh)

Iterasi 10

$$29 \rightarrow 30 = 7.705 + 175$$

$$= 7.880$$

$$29 \rightarrow 31 = 7.705 + 910$$

$$= 8.615$$

Adapun Hasil output menggunakan MATLAB adalah sebagai berikut:

t =

1

menghitung jarak terpendek untuk setiap titik dari titik asal

t =

1

Label(1) = [0 , 0]

Label(2) = [1 , 2100]

Label(3) = [2 , 3100]

Label(4) = [3 , 5960]

Label(5) = [4 , 6547]

Label(6) = [4 , 6520]

Label(7) = [4 , 6248]

Label(8) = [12 , 6795]

Label(9) = [3 , 3130]

Label(10) = [9 , 3505]

Label(11) = [10 , 4940]

Label(12) = [11 , 5675]

Label(13) = [12 , 6725]

Label(14) = [13 , 7495]

Label(15) = [14 , 7740]

Label(16) = [7 , 7053]

Label(17) = [15 , 9171]

Label(18) = [19 , 7775]

Label(19) = [21 , 7207]

Label(20) = [19 , 7642]

Label(21) = [11 , 6655]

Label(22) = [23 , 7075]

Label(23) = [11 , 6025]

Label(24) = [25 , 8405]

Label(25) = [23 , 6865]

Label(26) = [25 , 7110]

Label(27) = [26 , 8370]

Label(28) = [25 , 7135]

Label(29) = [28 , 7705]

Label(30) = [29 , 7880]

Label(31) = [29 , 8615]



B. Pembahasan

Berdasarkan hasil penelitian, terdapat 15 nama jalan yang diteliti di kecamatan Biringkanaya yang dituangkan dalam bentuk graf dimana terdapat 31 titik yang dicari rute terpendeknya. 31 titik tersebut merupakan batas-batas jalan. ini bisa diperhatikan pada Gambar 4.2 sebagai rute jalan kecamatan Biringkanaya.

Dari hasil penelitian di atas, untuk mencari rute terpendek melalui algoritma Bellman-Ford menggunakan dua cara yaitu menggunakan cara manual dan juga menggunakan bantuan program MATLAB. Penyelesaian pada kasus ini menggunakan algoritma Bellman-Ford menggunakan beberapa tahap. Namun sebelum menyelesaikan kasus ini terlebih dahulu perlu mencari jarak dari masing-masing titik. Pada penelitian ini menggunakan hasil pencarian dengan bantuan tool penunjuk arah pada google Maps.

Selanjutnya masuk kedalam algoritma pencarian jalur tependek dengan menggunakan Algoritma bellman-Ford. Tahap ke-1 adalah menentukan titik 1 sebagai titik awal dengan memberi simbol S dan mendaftar semua titik maupun sisi sebagaimana terdapat pada gambar 4.3. Setelah itu memberi nilai titik awal sama dengan nol dan yang lainnya tak terhingga, terdapat pada Gambar 4.4

Selanjutnya proses iterasi terhadap semua titik untuk menentukan jarak dari semua titik yang berhubungan dengan titik asal dengan cara:

Jika jarak V lebih besar dari jarak $U + \text{bobot } UV$ maka jarak V diisi dengan jarak $U + \text{bobot } UV$. dimana U = titik asal, V = titik tujuan, UV = sisi yang menghubungkan U dan V . Langkah ini dilakukan sehingga semua titik terkunjungi.

Berdasarkan hasil, pencarian rute minimum dari titik awal ke semua titik dengan menggunakan algoritma bellman-ford, terdapat delapan proses iterasi yaitu sebagai berikut:

1. Iterasi pertama

Iterasi pertama terdapat pada Gambar 4.5, dimulai dari titik yang berhubungan dengan titik awal, yaitu dari titik 1 ke titik 2. Pada iterasi pertama jarak untuk titik 2 diisi dengan bobot titik 1 ditambah dengan jarak dari titik 1 ke titik 2 yaitu $0 + 2.100 = 2.100$.

2. Iterasi kedua

Untuk iterasi kedua pada Gambar 4.6, yaitu iterasi dari titik 2 ke titik 3 diisi dengan jarak dari titik 1 ke titik 2 ditambah dengan jarak dari titik 2 ke titik 3 yaitu $2.100 + 1.000 = 3.100$

3. Iterasi ketiga

Iterasi ketiga yang terdapat pada Gambar 4.7, yaitu iterasi dari titik 3 ke titik 4 dan titik 9. Pada iterasi ketiga titik 4 diisi dengan jarak dari titik 1 ke titik 3 yang diperoleh pada iterasi kedua ditambah dengan jarak minimum dari titik 3 ke titik 4 yaitu $3.100 + 2.860 = 5.960$. Dan untuk titik 9 diisi dengan jarak dari titik 1 ke titik 3 ditambah dengan jarak minimum dari titik 3 ke titik 9 yaitu $3.100 + 30 = 3.130$.

4. Iterasi keempat

Iterasi keempat yang terdapat pada Gambar 4.8, yaitu iterasi dari titik 4 ke titik 5, 6, 7 dan titik 7, dan iterasi dari titik 9 ke titik 10. Untuk titik 5 diisi dengan jarak dari titik 1 ke titik 4 yang diperoleh pada iterasi ketiga ditambah dengan jarak minimum dari titik 4 ke titik 5 yaitu $5.960 + 587 = 6.547$, kemudian titik 6 diisi dengan jarak dari titik 1 ke titik 4 ditambah dengan jarak minimum dari titik 4 ke titik 6 yaitu $5.960 + 560 = 6.520$. Kemudian untuk titik 7 diisi dengan jarak dari titik 1 ke titik 7 ditambah dengan jarak minimum dari titik 4 ke titik 7 yaitu $5.960 + 288 = 6.248$. Selanjutnya, untuk titik 8 diisi dengan jarak dari titik 1 ke titik 4 ditambah dengan jarak minimum dari titik 4 ke titik 8 yaitu $5.960 + 2.735 = 8.695$. Kemudian untuk titik 10 diisi dengan jarak minimum dari titik 1 ke titik 9 ditambah dengan jarak minimum dari titik 9 ke titik 10 yaitu $3.130 + 375 = 3.505$.

5. Iterasi kelima

Iterasi kelima yang terdapat pada Gambar 4.9, yaitu iterasi dari titik 7 ke titik 13 dan 16, iterasi dari titik 8 ke titik 12, dan iterasi dari titik 10 ke titik 11. Untuk titik 13 diisi dengan jarak dari titik 1 ke titik 57 ditambah dengan jarak minimum dari titik 7 ke titik 13 yaitu $6.248 + 630 = 6.878$. Untuk titik 16 diisi dengan jarak dari titik 1 ke titik 7 ditambah dengan jarak minimum dari titik 7 ke titik 16 yaitu $6.248 + 805 = 7.053$. Kemudian untuk titik 12 diisi dengan jarak dari titik 1

ke titik 8 ditambah dengan jarak minimum dari titik 8 ke titik 12 yaitu $8.695 + 1.120 = 9.815$.

Selanjutnya untuk titik 11 diisi dengan jarak dari titik 1 ke titik 10 ditambah dengan jarak minimum dari titik 10 ke titik 11 yaitu $3.505 + 1.435 = 4.940$.

6. Iterasi keenam

Iterasi keenam terdapat pada Gambar 4.10 yaitu iterasi dari titik 11 ke titik 12, 21 dan 23, iterasi dari titik 12 ke titik 8 dan 13. Dan iterasi dari titik 13 ke 14. Untuk titik 12 sudah diisi pada iterasi kelima yaitu 9.815, akan tetapi pada iterasi keenam jarak yang diperoleh lebih minimum yaitu jarak dari titik 1 ke titik 11 ditambah dengan jarak minimum dari titik 11 ke titik 12 yaitu $4.940 + 735 = 5.675$. Karena 5.675 lebih kecil dari 9.815 maka titik 8 diisi dengan jarak 5.675. kemudian untuk titik 13 diisi dengan jarak dari titik 1 ke titik 12 ditambah dengan jarak minimum dari titik 12 ke titik 13 yaitu $5.675 + 1.050 = 6.725$. kemudian untuk titik 21 diisi dengan jarak dari 1 ke titik 11 ditambah dengan jarak minimum dari titik 11 ke titik 21 yaitu $4.940 + 1.085 = 6.655$. Kemudian untuk titik 23 diisi dengan jarak dari titik 1 ke titik 11 ditambah jarak minimum dari titik 11 ke titik 23 yaitu $4.940 + 1.085 = 6.025$.

Selanjutnya untuk titik 8 diisi dengan jarak dari titik 1 ke titik 12 ditambah dengan jarak minimum dari titik 12 ke titik 8 yaitu $5.675 + 1.120 = 6.795$. kemudian untuk titik 14 diisi dengan jarak dari titik 1 ke titik 13

ditambah dengan jarak minimum dari titik 13 ke titik 14 yaitu $6.725 + 770$
 $= 7.945$.

7. Iterasi ketujuh

Iterasi ketujuh yang terdapat pada gambar 4.11 yaitu iterasi dari titik 14 ke titik 15, iterasi dari titik 21 ke titik 19, iterasi dari titik 23 ke titik 22 dan 25. Untuk titik 15 diisi dengan jarak dari titik 1 ke titik 14 ditambah dengan jarak minimum dari titik 14 ke titik 15 yaitu $7.945 + 245$
 $= 7.740$

Selanjutnya untuk titik 19 diisi dengan jarak dari titik 1 ke titik 21 ditambah dengan jarak minimum dari titik 21 ke titik 19 yaitu $6.655 + 552$
 $= 7.207$. Kemudian untuk titik 22 diisi dengan jarak dari titik 1 ke titik 23 ditambah dengan jarak minimum dari titik 23 ke titik 22 yaitu $6.025 + 1.050 = 7.075$. Dan untuk titik 25 diisi dengan jarak dari titik 1 ke titik 23 ditambah dengan jarak minimum dari titik 23 ke titik 25 yaitu $6.025 + 840$
 $= 6.865$

8. Iterasi delapan

Iterasi delapan yang terdapat pada Gambar 4.12 yaitu iterasi dari titik 15 ke titik 17, iterasi dari titik 19 ke titik 18 dan 20. Iterasi dari titik 25 ke titik 24, 26 dan 28. Untuk titik 17 diisi dengan jarak dari titik 1 ke titik 15 ditambah dengan jarak minimum dari titik 15 ke titik 17 yaitu $7.740 + 1.431 = 9.171$, kemudian untuk titik 18 diisi dengan jarak dari titik 1 ke titik 19 ditambah dengan jarak minimum dari titik 19 ke titik 18 yaitu $7.207 + 435 = 7.642$, kemudian untuk titik 20 diisi dengan jarak dari

titik 1 ke titik 19 ditambah dengan jarak minimum dari titik 19 ke titik 20 yaitu 7.642.

Selanjutnya untuk titik 24 diisi dengan jarak dari titik 1 ke titik 25 ditambah dengan jarak minimum dari titik 25 ke titik 24 yaitu $6.865 + 1.540 = 8.405$. Kemudian untuk titik 26 diisi dengan jarak dari titik 1 ke titik 25 ditambah dengan jarak minimum dari titik 25 ke titik 26 yaitu $6.865 + 245 = 7.110$, kemudian untuk titik 28 diisi dengan jarak dari titik 1 ke titik 25 ditambah dengan jarak minimum dari titik 25 ke titik 28 yaitu $6.865 + 270 = 7.135$.

9. Iterasi kesembilan

Iterasi kesembilan yang terdapat pada gambar 4.13 yaitu iterasi dari titik 26 ke titik 27 dan iterasi dari titik 28 ke titik 29. Untuk titik 27 diisi dengan jarak dari titik 1 ke titik 26 ditambah dengan jarak minimum dari titik 26 ke titik 27 yaitu $7.110 + 260 = 8.370$

Selanjutnya untuk titik 29 diisi dengan jarak dari titik 1 ke titik 28 ditambah dengan jarak minimum dari titik 28 ke titik 29 yaitu $7.135 + 570 = 7.705$.

10. Iterasi kesepuluh

Iterasi ketujuh yang terdapat pada gambar 4.14 yaitu iterasi dari titik 29 ke titik 30 dan titik 31. Untuk titik 30 diisi dengan jarak dari titik 1 ke titik 29 ditambah dengan jarak minimum dari titik 29 ke titik 30 yaitu $7.705 + 175 = 7.880$. Selanjutnya untuk titik 31 diisi dengan jarak dari

titik 1 ke titik 29 ditambah dengan jarak minimum dari titik 29 ke titik 31 yaitu $7.705 + 910 = 8.615$.

Pada iterasi kesepuluh semua titik telah dikunjungi dan tidak ada lagi rute dengan jarak yang lebih minimum, maka proses iterasi selesai dan semua rute yang terdapat pada iterasi kesepuluh merupakan rute minimum. Berikut rute minimum dari titik 1 ke semua titik yang diperoleh dari proses iterasi.

1. Rute dari titik 1 ke titik 2

Dari titik 1 ke titik 2 hanya ada satu rute yang bisa dilewati sehingga dengan mudah diketahui rute minimumnya, namun berdasarkan algoritma bellman ford rute dari titik 1 ke titik diperoleh iterasi pertama yaitu dari titik $1 \rightarrow 2$ dengan jarak 2.100 meter yang terdapat pada gambar 4.5.

2. Rute dari titik 1 ke titik 3

Sama halnya titik 2, dari titik 1 ke titik 3 juga hanya terdapat 1 rute yang bisa dilewati yaitu titik $1 \rightarrow 2 \rightarrow 3$ dengan jarak 3.100 meter. Sehingga rute tersebut merupakan rute minimum. Ini bisa diperhatikan pada iterasi kedua yang terdapat pada Gambar 4.6.

3. Rute dari titik 1 ke titik 4

Dari titik 1 ke titik 4 ada beberapa jalur yang bisa dilewati dengan melewati titik 3, akan tetapi berdasarkan algoritma bellman-ford rute dari titik 1 ke titik 4 yaitu titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ dengan jarak

5.960 meter. Rute ini diperoleh pada iterasi ketiga yang terdapat pada Gambar 4.7.

4. Rute dari titik 1 ke titik 5

Diari titik 1 ke titik 5 juga terdapat beberapa rute yang bisa dilewati, namun adapun rute minimum yang diperoleh titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ dengan jarak 6.547 meter. Rute ini diperoleh pada iterasi keempat yang terdapat pada Gambar 4.8

5. Rute dari titik 1 ke titik 6

Sama halnya titik 4 dan titik 5, untuk sampai ke titik 6 juga ada beberapa rute yang bisa dilewati, namun rute yang paling minimum yaitu titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$ dengan jarak 6.520 meter. Berdasarkan algoritma bellman ford, rute tersebut diperoleh dari iterasi keempat yang terdapat pada Gambar 4.8.

6. Rute dari titik 1 ke titik 7

Adapun rute minimum dari titik 1 ke titik 7 yang diperoleh yaitu titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7$ dengan jarak 6.248 meter. Rute ini diperoleh pada iterasi keempat yang terdapat pada Gambar 4.8.

7. Dari titik 1 ke titik 8

Adapun rute minimum dari titik 1 ke titik 8 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 8$ dengan jarak 6.795. Rute ini diperoleh pada iterasi keenam yang terdapat pada Gambar 4.10

8. Rute dari titik 1 ke titik 9

Rute minimum dari titik 1 ke titik 9 diperoleh pada iterasi ketiga yang terdapat pada Gambar 4.7. Adapun rutenya yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9$ dengan jarak 3.130 meter.

9. Dari titik 1 ke titik 10

Rute minimum dari titik 1 ke titik 10 diperoleh pada iterasi keempat. Adapun Rutenya yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10$ dengan jarak 3.505 meter yang terdapat pada Gambar 4.8.

10. Rute dari titik 1 ke titik 11

Rute minimum dari titik 1 ke titik 11 diperoleh pada iterasi kelima Adapun Rutenya yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11$ dengan jarak 4.940 meter yang terapat pada Gambar 4.9.

11. Rute dari titik 1 ke titik 12

Rute minimum dari titik 1 ke titik 12 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12$ dengan jarak 5.675 meter. Rute ini diperoleh dari iterasi keenam yang terdapat pada Gambar 4.10.

12. Rute dari titik 1 ke titik 13

Rute minimum dari titik 1 ke titik 13 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13$ dengan jarak 6.725 meter. Rute ini diperoleh dari iterasi 6 yang terdapat pada Gambar 4.10.

13. Rute dari titik 1 ke titik 14

Rute minimum dari titik 1 ke titik 14 diperoleh dari iterasi keempat yaitu dari titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14$ dengan jarak 7.495 meter yang terdapat pada Gambar 4.10.

14. Rute dari titik 1 ke titik 15

Rute minimum dari titik 1 ke titik 15 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$ dengan jarak 7.740 meter. Rute ini diperoleh pada iterasi ketujuh yang terdapat pada Gambar 4.11.

15. Rute dari titik 1 ke titik 16

Rute minimum dari titik 1 ke titik 16 diperoleh dari iterasi kelima. Adapun rutenya yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 16$ dengan jarak 7.053 meter, terdapat pada Gambar 4.9.

16. Rute dari titik 1 ke titik 17

Rute minimum dari titik 1 ke titik 17 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11$ dengan jarak 9.171. Rute ini diperoleh dari iterasi kedelapan yang terdapat pada Gambar 4.12.

17. Rute dari titik 1 ke titik 18

Rute minimum dari titik 1 ke titik 18 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 21 \rightarrow 19 \rightarrow 18$ dengan jarak 7.775 meter. Rute ini diperoleh dari iterasi kedelapan yang terdapat pada Gambar 4.12

18. Rute dari titik 1 ke titik 19

Rute minimum dari titik 1 ke titik 19 diperoleh dari iterasi ketujuh yaitu dari titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 21 \rightarrow 19$ dengan jarak 7.207 meter yang terdapat pada Gambar 4.11.

19. Rute dari titik 1 ke titik 20

Rute minimum dari titik 1 ke titik 20 diperoleh dari iterasi kedelapan. Adapun rutennya yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 21 \rightarrow 19 \rightarrow 20$ dengan jarak 7.642 meter yang terdapat pada Gambar 4.8

20. Rute dari titik 1 ke titik 21

Rute minimum dari titik 1 ke titik 21 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 21$ dengan jarak 6.655 meter. Rute ini diperoleh dari iterasi keenam yang terdapat pada Gambar 4.10.

21. Rute dari titik 1 ke titik 22

Rute minimum dari titik 1 ke titik 22 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 22$ dengan jarak 7.075 meter. Rute ini diperoleh dari iterasi ketujuh yang terdapat pada Gambar 4.11.

22. Rute dari titik 1 ke titik 23

Rute minimum dari titik 1 ke titik 23 yaitu $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23$ dengan jarak 6.025 meter. Rute ini diperoleh dari iterasi keenam yang terdapat pada Gambar 4.10.

23. Rute dari titik 1 ke titik 24

Rute minimum dari titik 1 ke titik 24 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 24$ dengan jarak 8.405 meter. Rute ini diperoleh dari iterasi kedelapan yang terdapat pada Gambar 4.12

24. Dari titik 1 ke titik 25

Rute minimum dari titik 1 ke titik 25 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 15 \rightarrow 20 \rightarrow 23 \rightarrow 25$ dengan jarak 6.865 meter. Rute ini diperoleh dari iterasi ketujuh yang terdapat pada Gambar 4.11

25. Rute dari titik 1 ke titik 26

Rute minimum dari titik 1 ke titik 26 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 26$ dengan jarak 7.110 meter. Rute ini diperoleh dari iterasi kedelapan yang terdapat pada Gambar 4.12.

26. Rute dari titik 1 ke titik 27

Rute minimum dari titik 1 ke titik 27 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 26 \rightarrow 27$ dengan jarak 8.370 meter. Rute ini diperoleh pada iterasi kesembilan yang terdapat pada Gambar 4.13.

27. Rute dari titik 1 ke titik 28

Rute minimum dari titik 1 ke titik 28 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 28$ dengan jarak 7.135 meter. Rute ini diperoleh dari iterasi kedelapan yang terdapat pada Gambar 4.12.

28. Rute dari titik 1 ke titik 29

Rute minimum dari titik 1 ke titik 29 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 28 \rightarrow 29$ dengan jarak 7.705 meter.
Rute ini diperoleh dari iterasi kesembilan yang terdapat pada Gambar 4.13

29. Rute dari titik 1 ke titik 30

Rute minimum dari titik 1 ke titik 30 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 28 \rightarrow 29 \rightarrow 30$ dengan jarak 7.880 meter. Rute ini diperoleh dari iterasi kesepuluh yang terdapat pada Gambar 4.14

30. Rute dari titik 1 ke titik 31

Rute minimum dari titik 1 ke titik 31 yaitu
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 23 \rightarrow 25 \rightarrow 28 \rightarrow 29 \rightarrow 31$ dengan jarak 8.615 meter. Rute ini diperoleh dari iterasi kesepuluh yang terdapat pada Gambar 4.14

Selanjutnya berdasarkan output hasil program yang telah diperoleh menggunakan bantuan MATLAB yaitu menginput matriks 29×29 . Ini bersesuaian dengan banyaknya titik pada graf. Setelah menginput matriksnya, selanjutnya menentukan titik 1 sebagai titik awal yang disimbolkan dengan huruf t dan mendefinisikan semua titik. Setelah semua titik didefinisikan, langkah berikutnya adalah melakukan inisialisasi terhadap semua titik dimana titik awal diberikan nilai nol dan titik lainnya diberikan label 9999 angka 9999 digunakan sebagai pengganti

label tak hingga. Selanjutnya, menjalankan perulangan menggunakan algoritma Bellman-Ford dengan merelaksasi setiap sisi. Setelah itu program dijalankan maka diperoleh jarak tempuh minimum dari titik awal ke semua titik sebagai berikut:

t =

1

menghitung jarak terpendek untuk setiap titik dari titik asal

t =

1

Label(1) = [0 , 0]

Label(2) = [1 , 2100]

Label(3) = [2 , 3100]

Label(4) = [3 , 5960]

Label(5) = [4 , 6547]

Label(6) = [4 , 6520]

Label(7) = [4 , 6248]

Label(8) = [12 , 6795]

Label(9) = [3 , 3130]

Label(10) = [9 , 3505]

Label(11) = [10 , 4940]

Label(12) = [11 , 5675]

Label(13) = [12 , 6725]

Label(14) = [13 , 7495]

Label(15) = [14 , 7740]

Label(16) = [7 , 7053]

Label(17) = [15 , 9171]

Label(18) = [19 , 7775]

Label(19) = [21 , 7207]

Label(20) = [19 , 7642]

Label(21) = [11 , 6655]

Label(22) = [23 , 7075]

Label(23) = [11 , 6025]

Label(24) = [25 , 8405]

Label(25) = [23 , 6865]

Label(26) = [25 , 7110]



Label(27) = [26 , 8370]

Label(28) = [25 , 7135]

Label(29) = [28 , 7705]

Label(30) = [29 , 7880]

Label(31) = [29 , 8615]

>>



BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa pada penelitian ini terdapat 15 nama jalan yang diteliti yang dituangkan dalam bentuk graf. Dalam graf terdapat 31 titik yang dicari rute minimumnya, dimana 31 titik tersebut merupakan batas-batas jalan. Pada hasil penelitian, untuk menghitung rute terpendek dengan menggunakan Algoritma Bellman-Ford dilakukan dengan dua cara yaitu dengan cara manual dan dengan menggunakan program matlab.

Adapun rute minimum dari titik 1 ke semua titik yang diperoleh baik secara manual maupun menggunakan program Matlab yaitu:

Dari titik 1→2 sepanjang 2.100 meter, dari titik 1→3 sepanjang 3.100 meter, dari titik 1→4 sepanjang 5.960 meter, dari titik 1→5 sepanjang 6. 547 meter, dari titik 1→6 sepanjang 6.520 meter, dari titik 1→7 sepanjang 6.248 meter, dari titik 1→8 sepanjang 6.795 meter, dari titik 1→9 sepanjang 3.130 meter, dari titik 1→10 sepanjang 3.505 meter, dari titik 1→11 sepanjang 4.940 meter, dari titik 1→12 sepanjang 5.675 meter, dari titik 1→13 sepanjang 6.725 meter, dari titik 1→14 sepanjang 7.945 meter, dari titik 1→15 sepanjang 7.740 meter, dari titik 1→16 sepanjang 7.053 meter, dari titik 1→17 sepanjang 9.171 meter, dari titik 1→18 sepanjang 7.775 meter, dari titik 1→19 sepanjang 7.207 meter, dari titik 1→20 sepanjang 7.642 meter, dari titik 1→21 sepanjang 6.655 meter, dari titik 1→22 sepanjang 7.075 meter, dari titik 1→23 sepanjang 6.025 meter,

dari titik 1→24 sepanjang 8.405 meter, dari titik 1→25 sepanjang 6.865 meter, dari titik 1→26 sepanjang 7.110 meter, dari titik 1→27 sepanjang 8.370 meter, dari titik 1→28 sepanjang 7.135 meter, dari titik 1→29 sepanjang 7.705 meter, dari titik 1→30 sepanjang 7.880 meter, dari titik 1→31 sepanjang 8.615 meter.

B. Saran

Dalam penelitian ini yaitu mencari rute minimum perjalanan tukang Bentor yang ada di kecamatan Biringkanaya menggunakan Algoritma Bellman_Ford sehingga untuk penelitian berikutnya penulis menyarankan algoritma lain dan selanjutnya dapat dibandingkan hasil dari algoritma Bellman-Ford.

DAFTAR PUSTAKA

- Abidin, Wahyuni. "*Matematika Diskrit*". Makassar: Buku Daras UIN Makassar. 2013.
- Andana, Galih. "*Algoritma Bellman- Ford dalam Distance Vektor Routing Protocol*". <http://informatika.stei.itb.ac.id/rinaldi.munir/stmik/2009-2010/Makalah2009/MakalahIF3051-2009-044.pdf>. 2009. (25 Mei 2014)
- Anonim. <http://repository.usu.ac.id/bitstream/123456789/4236//Chapter%20I.pdf>
- Aprian, Raden Dias Novandi. "*Pebandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path)*". 2007.
- Ardiansyah, Irfan, dkk. "*Perancangan Aplikasi untuk Menentukan Jalur Terpendek Menggunakan Algoritma Floyd di wilayah Purbalingga*". 2012.
- Away, Gunaidi Abdia. "*The Shortcut of Matlab Programming*". Bandung: Informatika. 2006.
- Dany satrio kintono.blogspot.com. (29 januari 2015)
- Dedy Barnabas Lasfeto dan Oky Dwi Nurhayati. "*Analisis Statistika Deskriptif Menggunakan Matlab*". Yogyakarta: Graha Ilmu. 2008.
- Departemen Agama RI. "*Alqur'an dan Terjemahannya*". Jakarta: Perwakilan Bagian Percetakan dan Penerbitan Kementerian Agama. 2002.
- Drexel University. <http://mathforum.org/>. 199610 Juni 2014.
- Handika, Sri. "*Algoritma Bellman-Ford Sebagai Solusi Akses Tercepat dalam Jaringan computer*". 2009.
- Hartrand, lesneik. "*Matematika Diskrit*". Bandung: Erlangga. 1986
- <http://pahlawanbetopenk.blogspot.com/2011/01/matematika-diskrit.html>. (9 Juli 2014).
- http://yulieee.wordpress.com/2010/04/22/graf_tak_berarah/ (Online). (9 Juli 2014).
- Ibnu Katsir Ad-Dimayqi. "*Tafsir Ibnu Katsir Juz I*". Bandung: Penerbit Sinar baru Algensindo. 2000.
- Kamanyudi. "*Studi dan Implementasi algoritma Dijkstra, Bellman-Ford dan floyd- Warshall dalam Menangani Masalah Lintasan Terpendek Dalam*

Graf". <http://informatika.stei.itb.ac.id/rinaldi.munir/Matcis/2006-2007/Makalah/Makalah0607-115.pdf>. 2007. (10 Juni 2014).

Lipshutz, Seymour dan Mar Lars Lipson. "*Matematika Diskrit*". Jakarta: Salemba Teknik. 2002.

Munir, Rinaldi. "*Matematika Diskrit*". Diktat kuliah IF2091 Program Studi Informatika Institute Teknologi Bandung, http://id.wikipedia.org/wiki/Makalah_IF2091. 2009.

Nurholidah, Luluk. "*Keterhubungan Pada Graf Beraturan*". <http://lib.uin-malang.ac.id/thesis/fullchapter-luluk-nurholidah>. 2008.

Purwanto, Eko Budi. "*Perancangan dan Analisis Algoritma*". Yogyakarta: Graha Ilmu. 2008.

Shihab, M.Quraish. "*Tafsir Al-Misbah Pesan, Kesan dan Keserasian Al-Qur'an*". Vol.6; Jakarta: Lentera Hati. 2002.

The logo of Universitas Islam Negeri Alauddin Makassar is a green geometric emblem. It features a central yellow star with the year '1965' inside. The star is surrounded by concentric green arches and a larger green frame that resembles an open book or a stylized 'U' shape.

LAMPIRAN

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
M A K A S S A R

Lampiran 1:

Program Matlab Algoritma Bellman-Ford

```
clc
clear
fprintf('\n ALGORITMA Bellman-Ford Untuk mengetahui jalur terpendek \n');
ma = [
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	2100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2100	0	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	1000	0	2860	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	2860	0	587	560	288	2735	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	587	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	560	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	288	0	0	0	0	0	0	0	0	0	630	0	805	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	2735	0	0	0	0	125	0	0	1120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	375	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	1435	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	1435	0	735	0	0	0	0	0	0	0	1715	0	1085	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1120	0	0	735	0	1050	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	630	0	0	0	1050	0	770	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	770	0	245	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	245	0	700	1431	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	805	0	0	0	0	0	0	700	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1431	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	568	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2030	0	0	568	0	435	552	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	435	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	552	0	0	770	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	770	0	1050	0	0	840	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1050	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1540	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	245	0	1260	300	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1260	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270	300	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	570	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	175	910	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
]
tn=length(ma);
% menghitung jarak terpendek untuk semua titik dari titik asal
t=1
clearvars -except ma t tn jalur
m = ma;
% titik t sebagai titik asal
disp(['menghitung jarak terpendek untuk semua titik dari titik asal']);
mtb = m(1,:);
m(1,:) = m(t,:);
m(t,:) = mtb;
mtc = m(:,1);
m(:,1) = m(:,t);
m(:,t) = mtc;
t
```

```

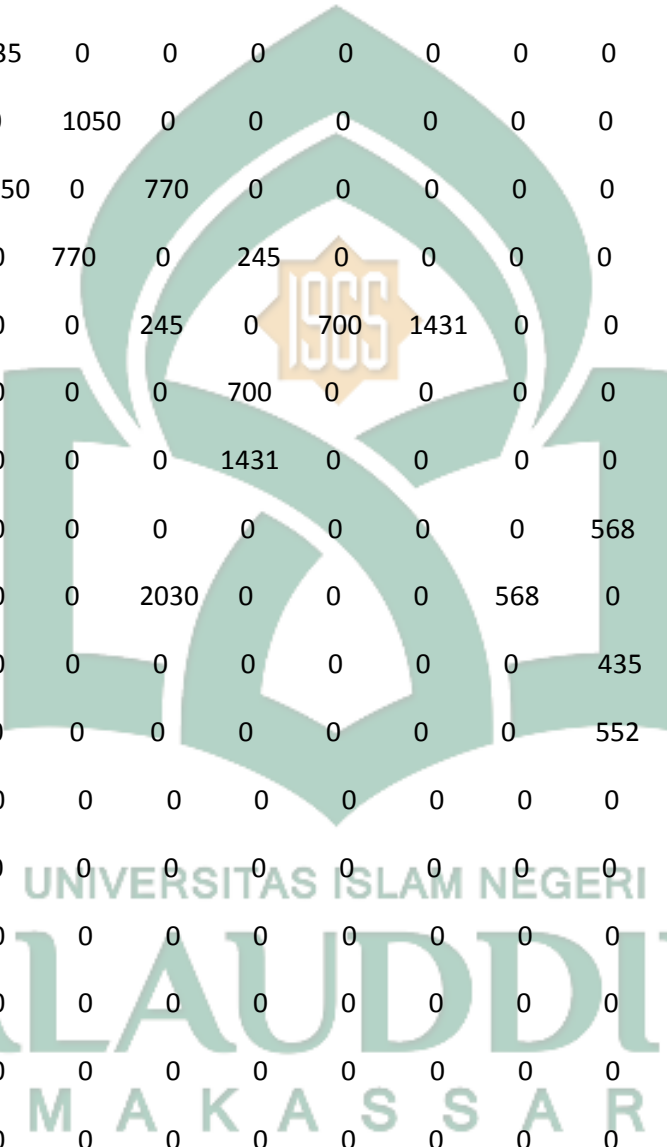
m;
k=1;
% mendefinisikan_semua_titik = [titik_pertama][titik_kedua]
for i=1:tn
    for j=1:tn
        if m(i,j) ~= 0 % jika sisi tidak ada maka dilangkahi
            edge(k,1) = i ;
            edge(k,2) = j;
            k=k+1;
        end
    end
end

% inisialisasi pertama semua titik diberikan nilai 999 mewakili tak hingga
% dan nol untuk titik awal
d(1)=0;
for i=2:tn
    d(i)=9999;
end

% sekarang kita jalankan perulangan menggunakan algoritma Bellman -Ford
% kita relaksasi untuk setiap sisi
% total banyaknya sisi= k-1
for i=1:tn
    for j=1:k-1
        if (d(edge(j,2)) > d(edge(j,1))+m(edge(j,1),edge(j,2)))
            d(edge(j,2)) = d(edge(j,1))+m(edge(j,1),edge(j,2));
            lastlabel(edge(j,2)) = edge(j,1);
        end
    end
end
% atur label

% if t>1
% tmp1 = d(t);
% d(t) = d(1);
% d(1) = tmp1;
for j=1:tn
    if lastlabel(j) == 1
        lastlabel(j)= t;
    elseif lastlabel(j) == t
        lastlabel(j)= 1;
    end
end
% printing shortest path
for i=1:tn
    if i==1
        ic = t;
    elseif i==t
        ic = 1;
    else
        ic = i;
    end
    jalur(ic, :, t)=[lastlabel(i) d(i)];
end

```

1435	0	0	0	0	0	0	0	0	0
0	735	0	0	0	0	0	0	0	0
735	0	1050	0	0	0	0	0	0	0
0	1050	0	770	0	0	0	0	0	0
0	0	770	0	245	0	0	0	0	0
0	0	0	245	0	700	1431	0	0	0
0	0	0	0	700	0	0	0	0	0
0	0	0	0	1431	0	0	0	0	0
0	0	0	0	0	0	0	0	568	0
0	0	0	2030	0	0	0	568	0	435
0	0	0	0	0	0	0	0	435	0
1715	0	0	0	0	0	0	0	552	0
0	0	0	0	0	0	0	0	0	0
1085	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Columns 21 through 30

0	840	0	0	245	0	1260	300	0	0
0	0	0	0	0	1260	0	0	0	0
0	0	0	0	270	300	0	0	570	0
0	0	0	0	0	0	0	570	0	175
0	0	0	0	0	0	0	0	175	0
0	0	0	0	0	0	0	0	910	0

Column 31

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

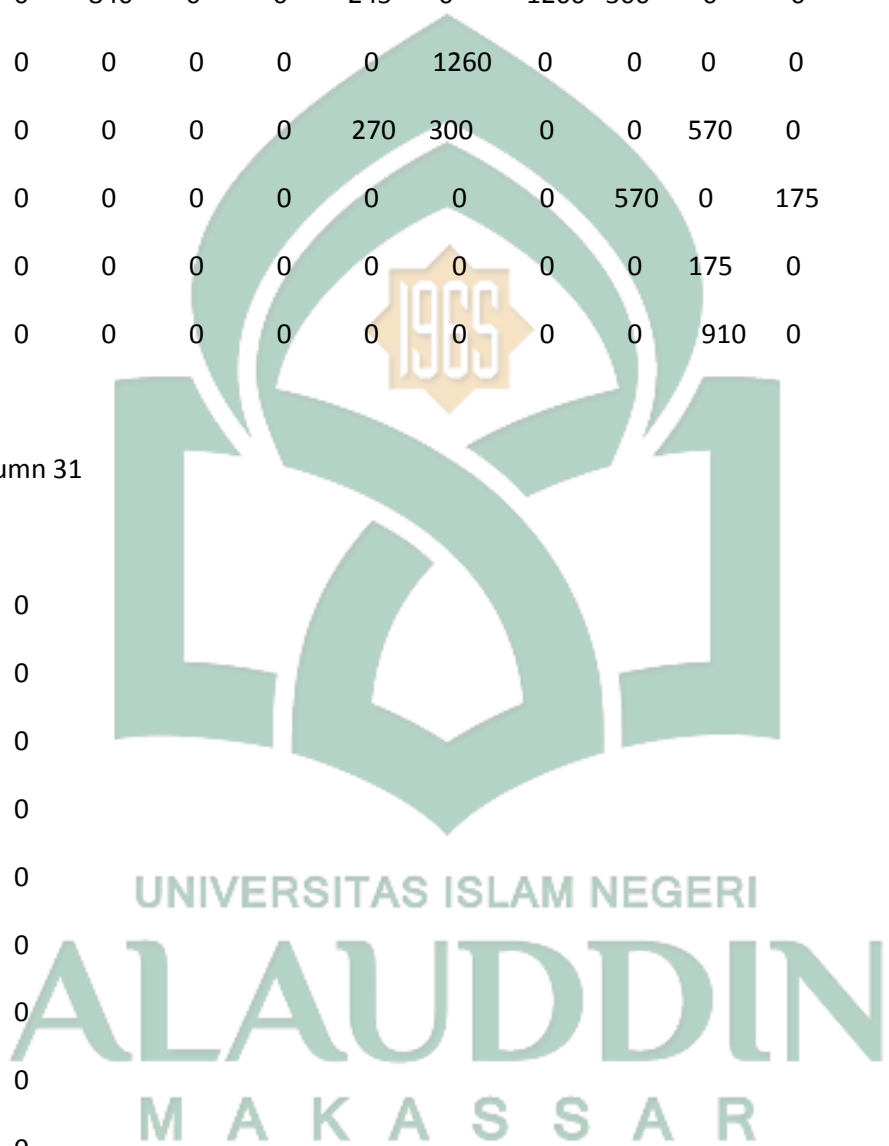
0

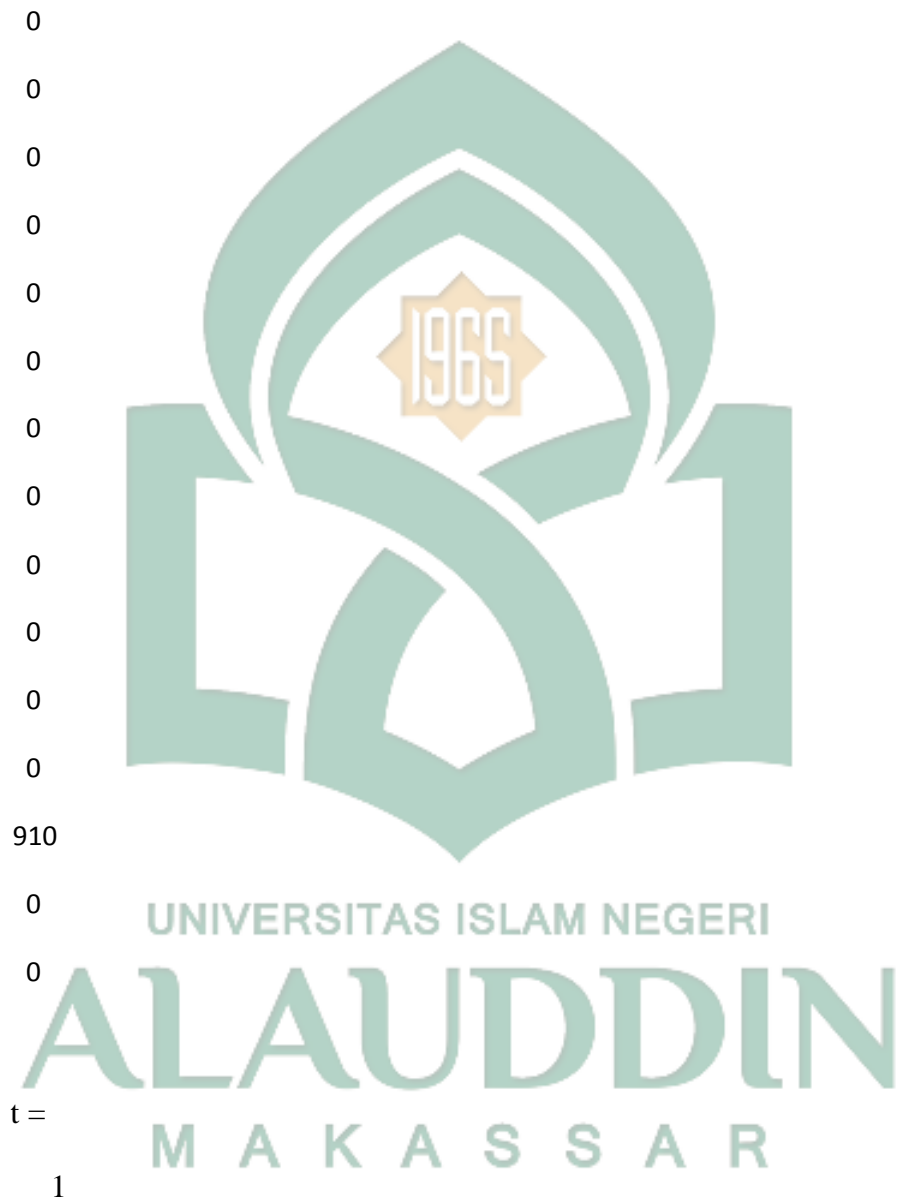
0

0

0

0





Label(4) = [3 , 5960]

Label(5) = [4 , 6547]

Label(6) = [4 , 6520]

Label(7) = [4 , 6248]

Label(8) = [12 , 6795]

Label(9) = [3 , 3130]

Label(10) = [9 , 3505]

Label(11) = [10 , 4940]

Label(12) = [11 , 5675]

Label(13) = [12 , 6725]

Label(14) = [13 , 7495]

Label(15) = [14 , 7740]

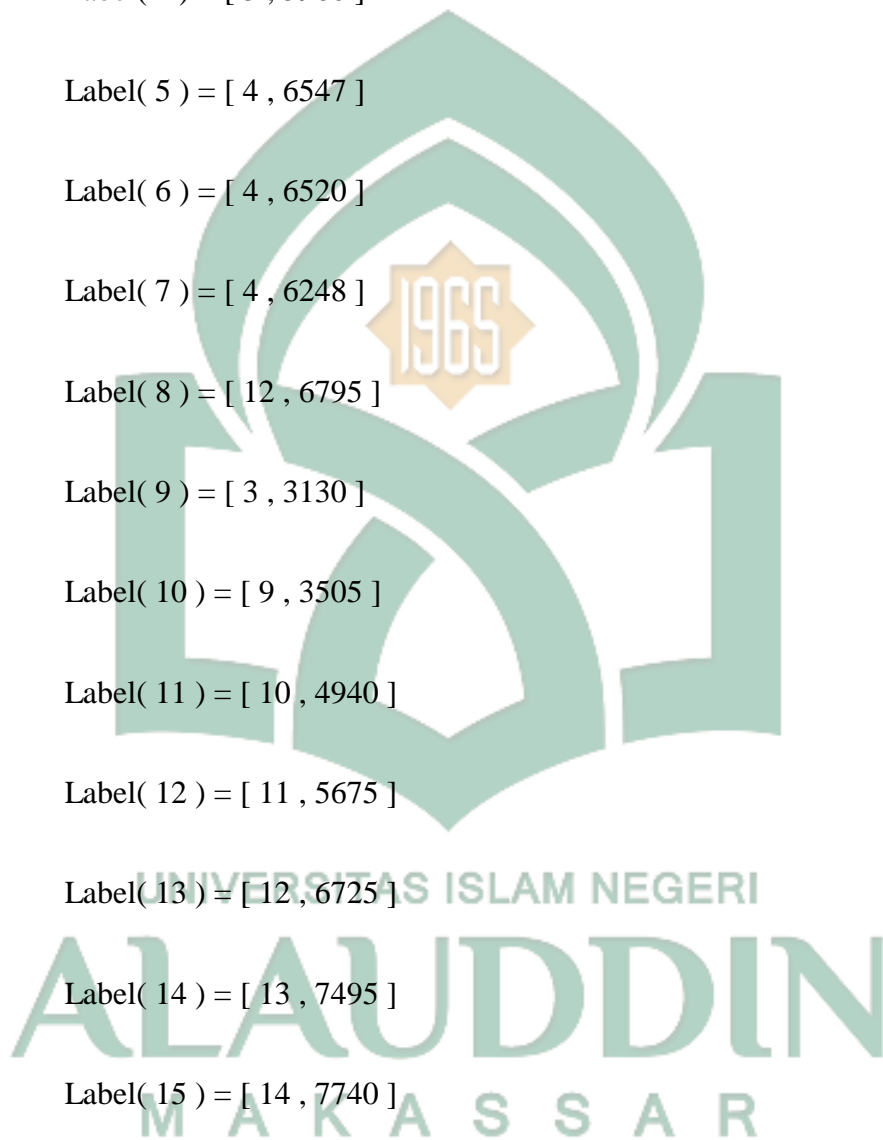
Label(16) = [7 , 7053]

Label(17) = [15 , 9171]

Label(18) = [19 , 7775]

Label(19) = [21 , 7207]

Label(20) = [19 , 7642]



Label(21) = [11 , 6655]

Label(22) = [23 , 7075]

Label(23) = [11 , 6025]

Label(24) = [25 , 8405]

Label(25) = [23 , 6865]

Label(26) = [25 , 7110]

Label(27) = [26 , 8370]

Label(28) = [25 , 7135]

Label(29) = [28 , 7705]

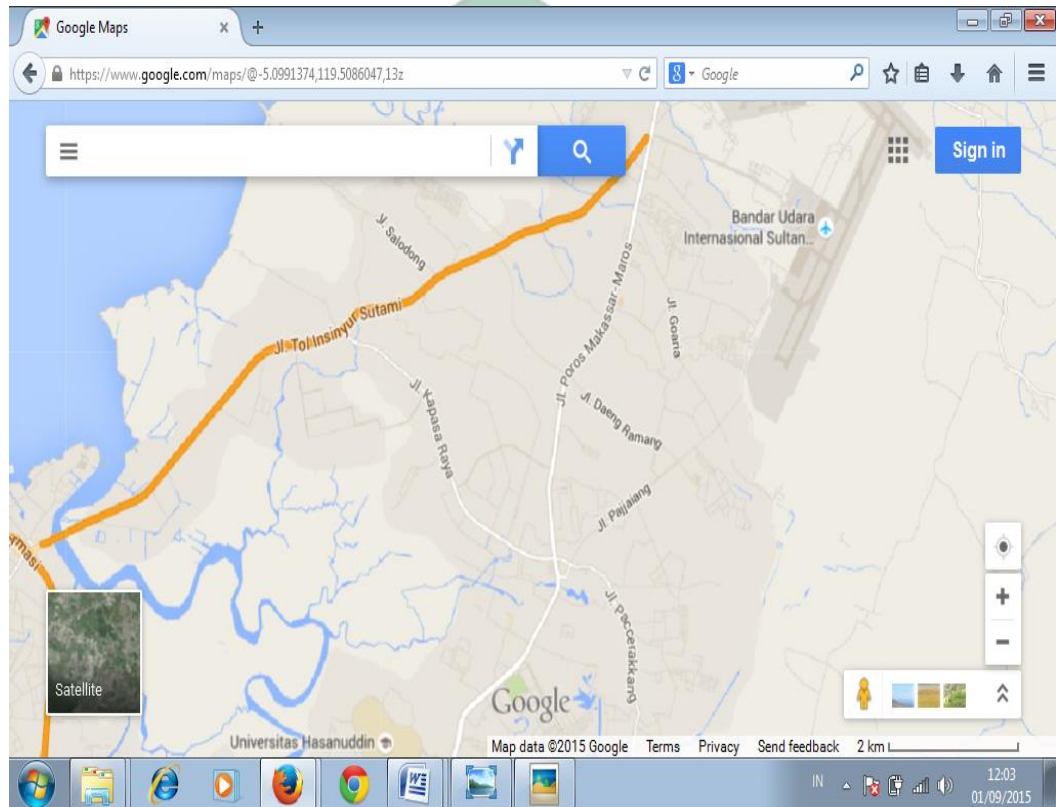
Label(30) = [29 , 7880]

Label(31) = [29 , 8615]

>>

M A K A S S A R

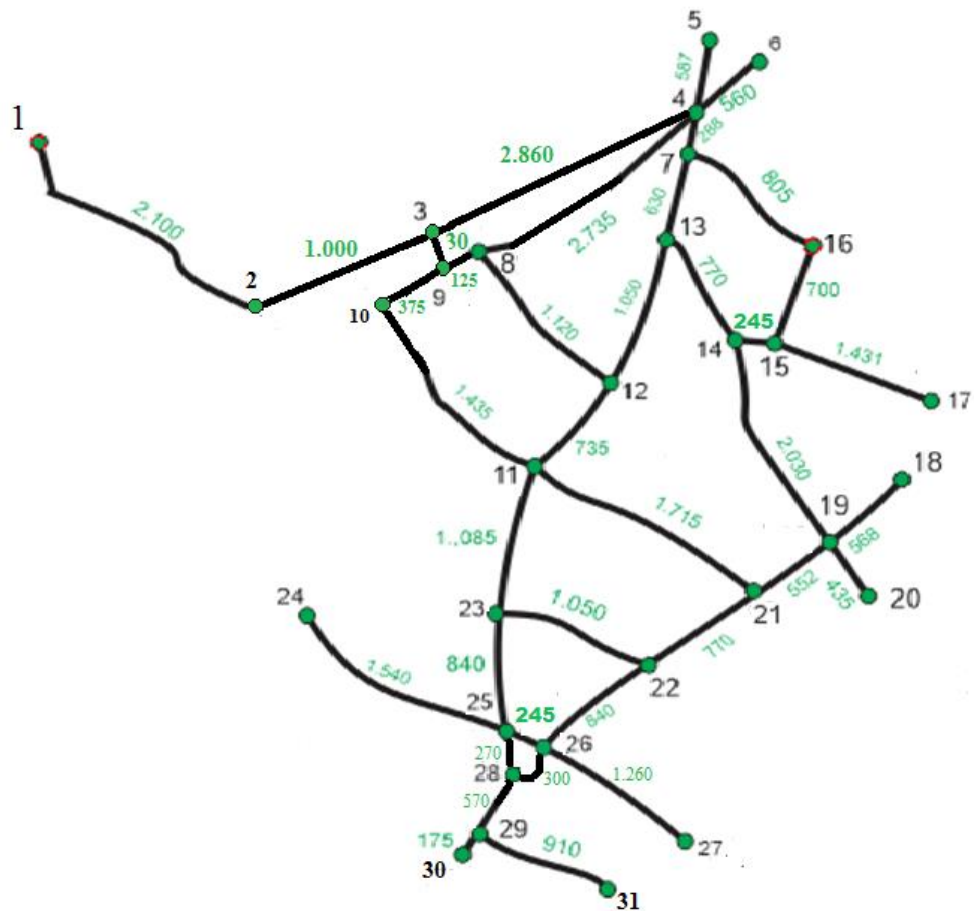
Lampiran 2 : Peta kecamatan Biringkanaya



Gambar Peta google Maps Kecamatan Biringkanaya

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
MAKASSAR

Lampiran 3



M A K A S S A R

Gambar Graf Kecamatan Biringkanaya

Keterangan:

1 - 2 : Jl. Salodong

2 - 3 - 4 - 5 - 7 : Jl. Ir. Sutami

3 - 15 : Jl Batara Bira

4 - 14 : Jl. H. Abd. Jabbar

6 - 5 - 8 - 10 - 14 - 15 - 20 - 23 - 26 - 27 - 28 : Jl. Perintis

Kemerdekaan

8 – 9 – 12 : Jl. Arung Teko

10 – 11 – 17 – 19 : Jl. Gowa ria

11 – 12 – 13 : Jl. Bakung

15 – 16 Jl. Dg. Rammang

17 – 18 Jl. Laikang

16 – 21 – 26 : Jl. Pajaiang B

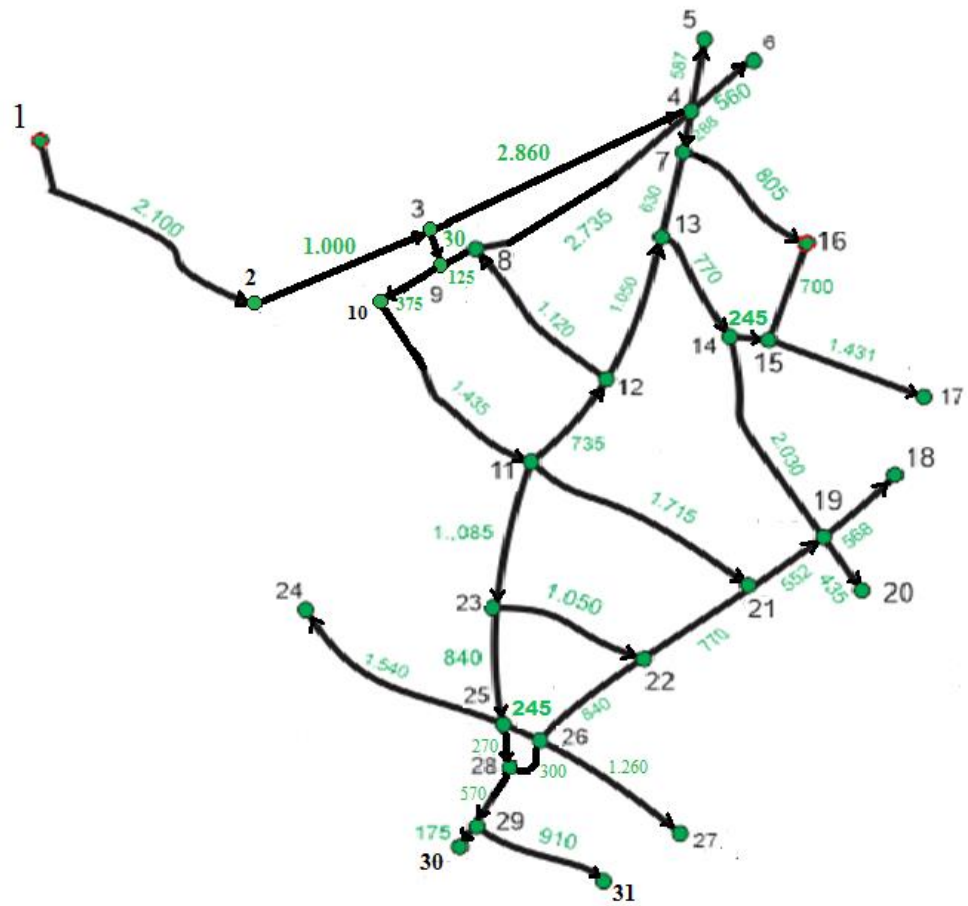
20 – 21 : Jl. Sanrangang

22 – 23 : Kapasa Raya

23 – 24 – 25 : Jl. Paccerakang

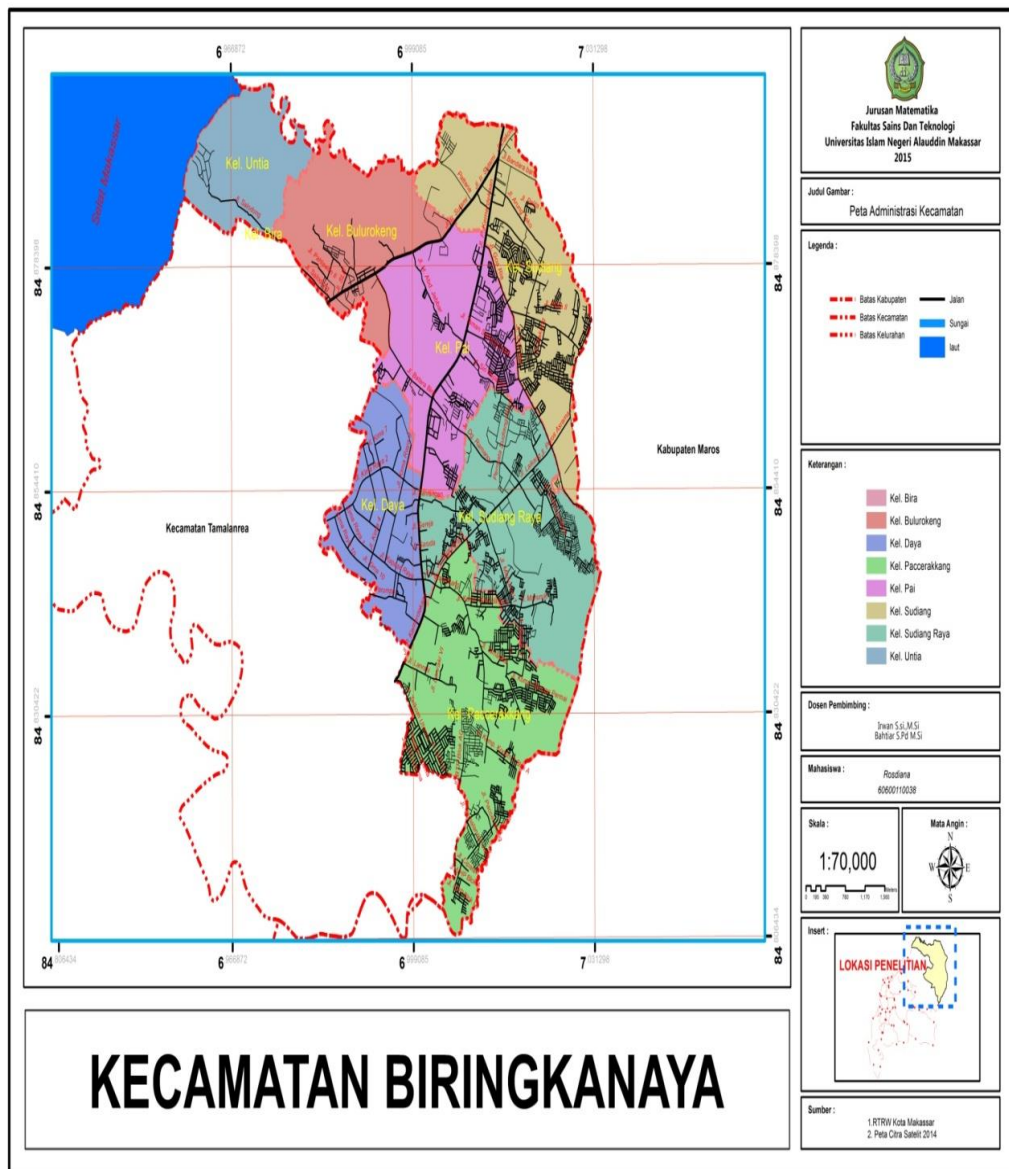
27 – 29 : Jl. Lanraki





Gambar graf Rute minimum dari titik 1 ke semua titik

ALAUDDIN
MAKASSAR





TIM VALIDASI PROGRAM STUDI MATEMATIKA
Fakultas Sains dan Teknologi
Universitas Islam Negeri Alauddin Makassar
Kampus II : Jalan Sultan Alauddin No. 36, Romang Polong, Gowa. Telp:(0411) 8221400

SURAT KETERANGAN
VALIDASI PENILAIAN KELAYAKAN DAN SUSBTANSI PROGRAM
No : 043 / Val / M / 358_2015

Yang bertanda tangan dibawah ini Tim Validasi penilaian kelayakan dan susbtansi program mahasiswa Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Alauddin Makassar menerangkan bahwa karya ilmiah mahasiswa :

Nama : **Rasdiana**
NIM : **60600110038**
Jurusan : **Matematika**
Judul Karya ilmiah

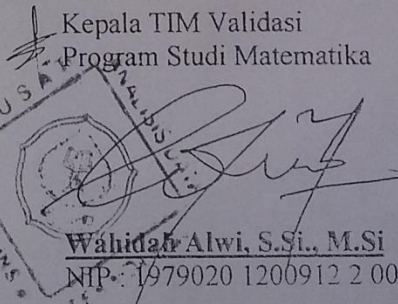
**“APLIKASI ALGORITMA BELLMAN-FORD DALAM
MEMINIMUMKAN RUTE PERJALANAN TUKANG BENTOR DI
KECAMATAN BIRINGKANAYA“**

Berdasarkan hasil penelitian kelayakan dan substansi program mahasiswa bersangkutan dengan ini dinyatakan Valid.

Demikian surat keterangan ini dibuat untuk digunakan sebagaimana mestinya.

Makassar, Agustus 2015

Kepala TIM Validasi
Program Studi Matematika


Wahidah Alwi, S.Si., M.Si
NIP. 1979020 1200912 2 002



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN) ALAUDDIN MAKASSAR
FAKULTAS SAINS & TEKNOLOGI
Kampus I: Jl. Sultan Alauddin No.63 Telp. 864924 (Fax 864923)
Kampus II: Jl. Sultan Alauddin No.36 Telp. 5622375-424835 (Fax 424836)

Nomor ST.VI.1/PP.003/29/2015
Sifat Penting
Lamp -
Hal Izin Penelitian
Untuk Menyusun Skripsi

Makassar, 31 Agustus 2015

Kepada Yth
Kepala Perpustakaan Umum
UIN Alauddin Makassar
Di-

Tempat

Assalamu Alaikum Wr. Wb.

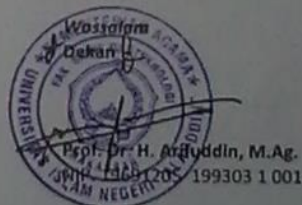
Dengan hormat kami sampaikan, bahwa mahasiswa UIN Alauddin Makassar yang tersebut namanya di bawah ini:

Nama	: Rasdiana
NIM	: 60600110038
Semester	: X
Fakultas	: Sains & Teknologi UIN Alauddin Makassar
Jurusan	: Matematika
Pembimbing	: 1. Irwan, S.Si., M.Si. 2. Bahtiar, S.Pd., M.Si.

Bermaksud melakukan penelitian dalam rangka penyusunan Skripsi berjudul " Algoritma Bellman-Ford dalam Meminimumkan Rute Perjalanan Tukang Bentor" sebagai salah satu syarat penyelesaian Studi akhir Sarjana/S 1.

Untuk maksud tersebut kami mengharapkan kiranya kepada mahasiswa yang bersangkutan diberi izin untuk penelitian di Perpustakaan Umum UIN Alauddin Makassar

Demikian harapan kami, atas perhatian dan kerjasamanya kami ucapkan terima kasih.



Tembusan:

1. Ketua Prodi/Jurusan Matematika Fak. Sainstek UIN Alauddin
2. Arsip



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI "ALAUDDIN" MAKASSAR
UPT PUSAT PERPUSTAKAAN

Jln. Sultan Alauddin No. 63 Telp. 864928-864931 (Fax 864923)

SURAT KETERANGAN
NO: PK/HM.02.1/ 48 /2015

Yang bertanda tangan di bawah ini, menerangkan bahwa :

Nama	: Rasdiana
Nim	: 606001110038
Semester	: X (Sepuluh)
Fakultas	: Sains & Teknologi UIN Alauddin Makassar
Jurusan	: Matematika

Yang bersangkutan telah melakukan izin penelitian pada tanggal 26 Januari s.d 26 Februari 2015 dengan Judul :

"Algoritma Bellman-Ford dalam Meminimumkan Rute Perjalanan Tukang Bentor"
di UPT Pusat Perpustakaan UIN Alauddin Makassar"

Demikian Surat Keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Samata, 7 September 2015



Kepala UPT Pusat Perpustakaan

S. Ag., SS., MIMS,
19730119 200003 2 002

RIWAYAT HIDUP



Rasdiana lahir di Pangkep pada tanggal 25 Juni 1992. Penulis merupakan anak ketiga dari lima bersaudara dari bapak Hambali dan ibu Husna. Penulis memulai pendidikan jenjang Sekolah Dasar (SD) di SDN 11 Ale Bonto-bonto pada tahun 1998, kemudian lanjut Sekolah Menengah Pertama (SMP) di SMPN 1 Salomekko pada tahun 2004, kemudian lanjut SMA di SMAN 1 Tomra pada tahun 2007 dan tamat pada tahun 2010 dan melanjutkan pendidikan jenjang S-1 di Universitas Islam Negeri (UIN) Alauddin Makassar pada Fakultas Sains dan Teknologi jurusan Matematika pada tahun 2010.

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
MAKASSAR